



СОДЕРЖАНИЕ

4	Notes Новости сообщества Open Source
10	Success story Новая жизнь старых компьютеров Настройка бездисковых станций под управлением LTSP
18	Desktop Системный обзор Сравнительный обзор современных дистрибутивов
24	Домашняя видеостудия Обзор программ для цифровой обработки видео в Linux
30	...плюс телефонизация всей Linux Практические советы и полезные утилиты для работы с сотовыми телефонами
33	Клеим панораму Работа с панорамными фотографиями в программе hugin
36	Плагиномания Улучшение Firefox и Thunderbird с помощью плагинов
41	Контора пишет... считает и рисует Новые возможности OpenOffice.org. 2.0
46	На ринге легковесы Небольшие по объему оконные менеджеры на любой вкус
56	Advanced Linux? Вам какой? Ядро Linux: особенности веток, рекомендации по выбору
62	Оконный дуэт Работа с X-серверами для Windows
66	Системный анабиоз «Усыпление» компьютера с помощью swsusp2
68	Раздолье в серверном заповеднике Виртуальные серверы — лаборатория администратора
73	Udev в разрезе Преимущества технологии для подключения устройств
76	Coding Анютины грезы Блеск и нищета среды разработки Anjuta
78	Полное затмение Платформа для Java-разработчиков и не только
82	Рабочая лошадка администратора Решение практических задач с помощью Python
88	Practice Фактический материал Полезные советы по решению проблем и тюнингу системы
94	Review Обзор интересных программ
96	Новинки книжного рынка

Ритм жизни

Минувшая весна и начало лета оказались богатыми на события в мире IT и Open Source. В апреле в Москве успешно прошел Open Source Forum Russia. Его посетили с докладами такие гуру как Джон «Мэддог» Холл и Лари Уолл, в непринужденной атмосфере кулуаров общались преподаватели и их студенты, разработчики и бизнесмены. Знаковым выглядит участие в конференции членов Правительства РФ, и хочется верить, что наше государство будет уделять больше внимания проектам с открытым кодом, как, например, Куба или в Южная Корея. Если вы не смогли посетить форум, то в нашем календаре событий сможете узнать о запланированных мероприятиях, в той или иной степени касающихся Linux. Совершенно неожиданным для публики стало заявление Apple о переходе на платформу Intel. Какой шаг предпримет теперь Линус Торвалдс? Как вы помните, недавно он мигрировал на Macintosh. Было это недальновидным поступком, или перспективы в этом направлении еще есть? Ведь и вышедший 13 мая дистрибутив Fedora Core 4 тоже поддерживает процессоры PowerPC. Впрочем, для большинства из нас установка Linux на Macintosh всего лишь интересная теоретическая возможность, а вот узнать о производительности, удобствах и недостатках популярных дистрибутивов было бы интересно. Составить собственное мнение о Mandrake, Fedora Core, Slackware и Gentoo вам поможет статья «Системный обзор». Компания Google не перестает удивлять своей креативностью. Тем, кто этого еще не сделал, настоятельно рекомендую заглянуть на страницу maps.google.com и получить фантастическую возможность заглянуть в любую точку земного шара. Но это еще не все, Google открыла программу поощрения студентов, пишущих ПО с открытым кодом, — Summer of Code. Программированию в этом номере посвящены целых три материала, рассказывающих о средах разработки (Anjuta и Eclipse) и практическом кодировании на языке Python. На диске вы найдете набор программ для работы, программирования и отдыха. Мы постарались приложить все необходимые для установки библиотеки, а также, где это было возможно, исходные коды. Оставайтесь всегда на острие событий, а мы поможем вам в этом!

Денис Михайлов, выпускающий редактор

Все лучшее — детям!

«Образование для всех». «Свободный обмен идеями и знаниями». «Повышенные значения коммуникации между людьми». Тезисы? Нет, руководство к действию. Так запечатлели для себя эти положения OFSET (организация по разработке свободного ПО для образования и обучения) и ЮНЕСКО в октябре 2003 года.



Несомненные достоинства открытого подхода к разработке софта нашли себя там, где и должны быть реализованы по своему определению. Результат не заставил себя долго ждать — первенцем стал Freeduc-cd 1.4, а недавно появилась уже и версия 1.5. Главный адресат Freeduc — средняя школа. Основная идея — организовать эффективную систему образования на основе свободного ПО: школе нужны не только офисные

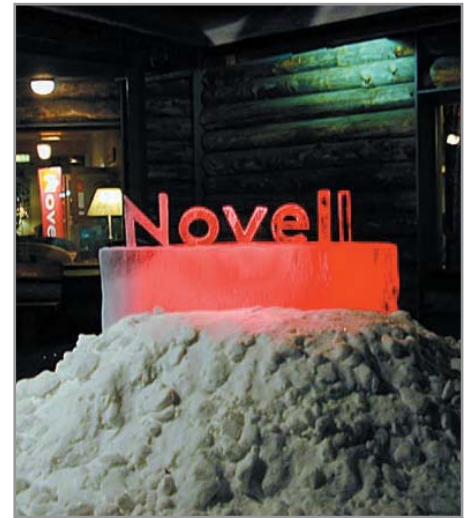
и коммуникационные, но и специфические приложения, отбираемые по критериям педагогики и эргономики, для поддержки процесса обучения. Добавьте сюда исходники, романтику совместной разработки — и детям! Дистрибутив выполнен в ставшем уже привычным виде LiveCD (основан на

KNOPPIX/Debian), что создает дополнительные удобства при использовании в школе, освобождая учителей от необходимости тратить время на установку и настройку системы. За организацию рабочего места отвечает элегантный Xfce, в качестве файлового менеджера используется ROX-Filer, в комплект входят также два диска с исходными кодами ПО.

www.ofset.org/freeduc-cd

Novell и Immunix

Компания Novell приобрела Immunix Inc., известную своими приложениями по безопасности для Linux и системой AppArmor. Финансовая сторона сделки не разглашается, но уже стало известно о запуске нового продукта, получившего в названии соответствующую приставку — Novell AppArmor. Он призван защитить саму ОС Linux и ее приложения от внешних и внутренних атак, вирусов и программ-злоумышленников.



Разделяй и властвуй!

Активности некоторых компаний можно позавидовать. Nokia продолжает удивлять нас: едва успев представить в Нью-Йорке на LinuxWorld Summit интернет-планшет Nokia 770 Internet Tablet, компания объявила о желании вложить средства, вырученные от продажи 500 моделей новых устройств, в GNOME Foundation, что стало приятным сюрпризом на очередной европей-

ской конференции разработчиков и пользователей GNOME (GUADec) в Штутгарте. Интерфейс Nokia 770 Internet Tablet основан на среде Hildon Application Framework, базирующейся на технологиях GNOME. Сам интернет-планшет имеет 4,13-дюймовый экран с разрешением 800x480 точек (65 536 цветов), включает экранную клавиатуру, работает с рукописным вводом, поддерживает масштабирование и полноэкранный режим работы. Напомним, что чуть раньше Nokia решила передать в потенциальное использование около четырех тысяч патентов в ядре Linux. При этом, правда, компания не отказалась от своей позиции, лоббирующей ужесточение патентного законодательства в Европе.

www.nokia.com; <http://2005.guadec.org>



Linux на Кубе

Недавно правительство Кубы решило отказаться от продуктов Microsoft в пользу Linux. Факторов, определяющих этот выбор, немало: безопасность, открытость, возможность участвовать в разработке необходимого ПО и даже политические соображения. Последние, видимо, сыграли немаловажную роль в принятии такого решения. Чрезвычайно привлекательной в отношении принципов государственной безопасности остается идея использовать собственный национальный дистрибутив, чего можно добиться с помощью Linux. К тому же это предотвращает отток финансовых средств из страны, создает благоприятные условия для работы талантливой молодежи, а конечный продукт вполне может стать предметом национальной гордости.

www.linuxinsider.com

Встроенные Linux-решения: примеры успеха

ARTS — многоцелевая дистанционная транспортная система, построенная на технологиях ОС реального времени LynxOS и управляемая специальным узлом контроля. Применяемая в условиях повышенного риска, она позволяет полностью контролировать работу машины на безопасном расстоянии. Подобные спецсредства активно используются при борьбе с пожарами и последствиями стихийных бедствий. Система уже имеет успешный опыт применения в операциях по разминированию в Афганистане. LynxOS также выбрана для применения в Future Combat Systems (FCS, Боевые системы будущего) армейской программы Integrated Computer System. Продукт от LynuxWorks будет использоваться в сете-

вых военно-воздушных и наземных системах. «Этот контракт отражает знаменательный этап не только для нашей компании, но и для всего движения встраиваемого ПО открытого стандарта», — подчеркивают представители LynuxWorks. www.lynxworks.com



Клятва Гиппократа

Министерство здравоохранения и социальных служб США с одной стороны и Novell с другой подписали соглашение о неограниченном использовании продуктов компании в системе здравоохранения США. Соглашение является составной частью инициативы по стимулированию закупок и распространению корпоративного программного обеспечения, поддерживающей программу президента США по перестройке федерального правительства. Результатом соглашения становится значительное снижение расходов и облегчение закупок ПО. С учетом охвата еще и национальных институтов здравоохранения Novell получает около 100 тысяч пользователей. <http://linux.slashdot.org>

Увертюра на тему Linux



Симбиоз KNOPPIX/Debian дал толчок построению новых дистрибутивов, позволяя разработчикам воплощать самые смелые замыслы, но идею этого дистрибутива можно в некотором смысле назвать и радикальной. Создатели Symphony решили в корне пересмотреть глобальный аспект построения — зависимости пакетов, представив авторам стороннего ПО полностью стандартизи-

рованную систему управления, вынуждающую предлагать для установки самодостаточные пакеты, содержащие все необходимые для работы приложения библиотеки и файлы. Этим дело не ограничивается. Symphony предоставляет возможность ваять шедевры графических интерфейсов с помощью программного окружения Orchestra так же легко, как разработчики пишут обычно скрипты Perl CGI. Мало того, вы не найдете здесь ни KDE, ни GNOME, ни даже Xfce — отказавшись от их использования, разработчики Symphony пошли навстречу владельцам маломощных компьютеров. Применяется простой FVWM, что позволяет относительно легко согласовать и общую концепцию построения системы на Orchestra/Perl. Наконец, главная изюминка дистрибутива с точки зрения ко-

нечного пользователя — Mezzo, воплощение «Законов разработки интерфейса» Джейсона Списака на рабочем столе пользователя. Отвергнув традиционный принцип «рабочий стол — это папка», привычную систему панелей и выпадающих меню, Mezzo предлагает решение, при котором в углах рабочего стола размещаются элементы быстрого доступа к полноэкранному меню настроек системы, приложениям, домашним директориям и корзине. При этом никакого скроллинга — все масштабируется автоматически. Удобство работы с настройками самой системы обеспечивается прямым доступом к ним — никаких лишних движений, никакого Drag&Drop. Дистрибутив рассчитан на домашних пользователей и мелких бизнесменов, текущее состояние — пререлиз www.symphonyos.com



Fedora — раз, два, три...

Четыре. Под именем Stentz вышел четвертый дистрибутив Fedora Core. Каждый из предыдущих в свое время становился настоящим событием в мире Linux. Что на этот раз? В первую очередь, и сами разработчики обращают на это внимание, — новое руководство по установке, одинаково полезное и для начинающих, и для опытных пользователей. И далее по порядку — ядро 2.6.11. Fedora стала первым дистрибутивом, собранным с помощью четвертого компилятора GNU C (GCC), включенного в систему по умолчанию. Рабочее окружение — GNOME 2.10 (см. статью «Праздник в подземном царстве» предыдущего номера). KDE 3.4.0 — еще «дышащая краской сборочного цеха». OpenOffice.org 1.9.104 — один из релизов, предвещающих выход окончательной версии 2.0. Xen 2 — средство виртуализации, позволяющее запускать одновременно несколько операционных систем. Кластерная файловая система — GFS 6.1-0.pre22. Универсальный просмотрщик документов — Evince 0.2.1.



Менеджер входа в систему — GDM 2.6. Улучшенный SELinux, теперь покрывающий еще 80 новых демонов. Eclipse 3.1M6 — полнофункциональная среда разработки на Java, Java-компилятор GCJ и новые возможности технологий Java. Теперь в Fedora Core вы не найдете ни Abiword, ни Gnumeric, ни даже KOffice. В составе дистрибутива нет также exim, XFce 4, Xemacs, xmms. Все это можно получить с помощью yum из Fedora Extras. Такое решение было принято исходя из принципа комплектации Core по возможности

единственными и самыми достойными представителями приложений одного функционального класса. Так, место офисных приложений занимает OpenOffice.org, агента доставки почтовых сообщений — Postfix, возможности Xfce с лихвой покрывают GNOME и KDE, Xemacs — Emacs, а xmms — Rhythmbox. Такое решение, с другой стороны, позволило вместить Fedora Core 4 на обычный комплект из четырех компакт-дисков.

Linux на рынке БД

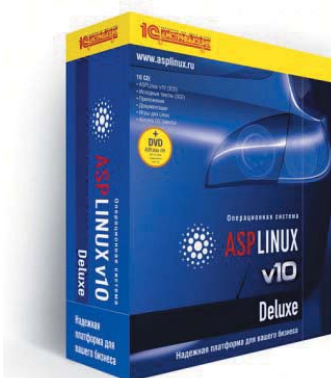
Рост числа корпоративных баз данных, работающих на платформе открытой ОС, продолжается. Как сообщает исследовательский центр Gartner, число лицензий для таких систем управления реляционными базами данных, собранных под Linux, как Sybase, IBM DB2 и Oracle, отражается сегодня в сумме \$654,8 млн. Хотя общий рост, по оценкам Gartner, за истекший год и достиг 118% на 7,7-миллиардном рынке, стоит отметить, что исследовательский центр не брал в расчет открытые продукты, такие как MySQL, Firebird и PostgreSQL. www.networkworld.com

Linux на PlayStation 3

Sony предполагает возможность поставки готовящейся к выходу новой приставки, имеющей дополнительный жесткий диск с предустановленной ОС Linux. Этот шаг должен, по мнению компании, наконец убедить общественность в том, что устройство все же имеет полное право называться компьютером. www.anandtech.com



Разработчики отечественного дистрибутива ASPLinux идут навстречу пожеланиям пользователей и предлагают обновление для ASPLinux 10 на двух компакт-дисках. В его



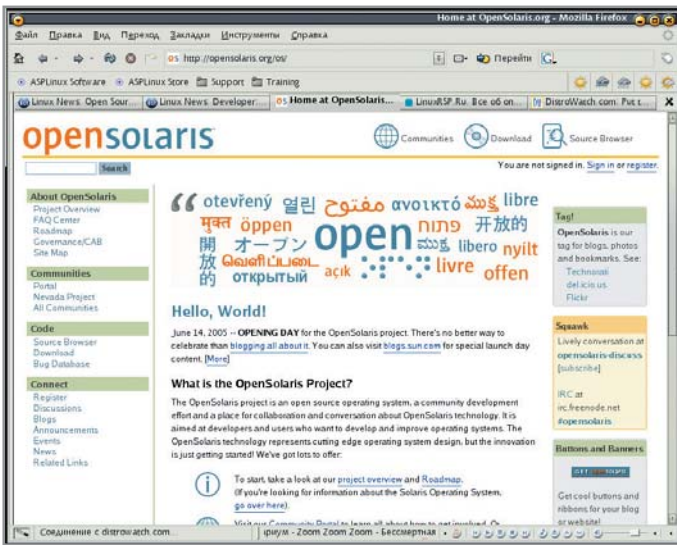
Преобразование ASPLinux

состав вошли ядро 2.6.11, свежая версия X.org X11 6.8.2, интегрированная среда GNOME 2.10.1, графическая среда Xfce 4.2.1, обозреватель Firefox 1.0.4 и почтовый клиент Thunderbird 1.0.2, эмулятор wine 20050524, антивирус ClamAV 0.85 и его графическая оболочка Klamav, графический редактор GIMP 2.2.6 — всего 740 бинарных пакетов.

Видимо, можно ожидать, что такая практика станет традиционной, а пользователи, вне зависимости от имеющегося доступа в Интернет, смогут и впредь получать обновления в наиболее приемлемой для себя форме. Возможно, свою роль в принятии

такого решения сыграла практика другого ведущего отечественного разработчика, ALT Linux, регулярно выпускающего срезы репозитория (банка пакетов) «Сизиф» на DVD, содержащие самое свежее ПО для своего дистрибутива. Несмотря на то что «Сизиф» напрямую не предназначен для обновления системы, это не мешает самым смелым пользователям устанавливать и использовать новые версии приложений. Если отправной точкой для «Сизифа» является разработка, то для предлагаемого ASPLinux комплекта в роли нее выступают непосредственно обновление и, как следствие, стабильность. www.asplinux.ru

Через тернии к звездам



Под специально созданной лицензией Common Development and Distribution Licence (CDDL) компания Sun открыла исходники своей ОС Solaris 10 и официально сообщила об этом в виде старта проекта OpenSolaris. Скачать исходники можно с официальной страницы нового проекта, собрать — с помощью gcc или Sun Compiler. Кроме того, любой желающий сегодня может ознакомиться с LiveCD, основанным на OpenSolaris Schillix, с которым можно работать как в обычном LiveCD-режиме, так и установив на жесткий диск или USB-флеш. Незадолго до этого события масла в огонь подлила NVIDIA, выпустив свежий драйвер для OpenSolaris с поддержкой OpenGL 2.0, Xinerama и собственной системы настроек. Перед объявлением о запуске проекта произошло еще одно интересное событие. Ватикан в выборе коммуникационных технологий остановился именно на Solaris и Java Enterprise System от Sun. Сдается, одно только название уже в определенной степени предопределило судьбу этого продукта. <http://opensolaris.org>

Будущее сотовой связи и Linux

Исследования ABI говорят об исключительной перспективности союза Linux со стандартами голосовой Wi-Fi-связи (VoWiFi). «VoWiFi — будущее мобильной телефонии», — говорится в большом докладе, где в расчет были взяты не только сами технологии и компании, их представляющие, но и показатели прогнозов рынка до 2010 года. Глава ABI Research Филип Солис пишет, что и внедряемая компанией British Telecom технология Bluephone, выбравшая интерфейс Bluetooth для сервиса VoIP, — толь-

ко промежуточный шаг, предтеча полноценного VoWiFi-решения. Соответствующие телефоны под управлением Linux уже производят Siemens и NexGen, а универсальные аппараты Mobile/VoWiFi представлены компаниями NEC (на фото справа — модель n900iL) и Panasonic. В свою очередь, такие компании как Motorola и Fujitsu оснащают аналогичные продукты решениями, основанными на Windows-разработках. www.abiresearch.com



Очередное приобретение Mandriva

Не успели остыть страсти вокруг приобретения теперь уже бывшего Mandrakesoft Connectiva и смены его имени, как общественности было объявлено о новой сделке: Mandriva приобретает значительную часть компании Lycoris. Последняя известна как крупнейший сборщик дистрибутивов для домашнего использования в Северной Америке Desktop/LX и начинатель на массовом рынке ОС Linux в Tablet Edition и Pocket PC Edition. Успехи Lycoris отразились в установлении

партнерства с азиатскими производителями оборудования — такими как Dialogue и MoBits. Интерес Mandriva к этой компании подогревает изначальной ориентированностью дистрибутивов Mandrake на дружелюбность к пользователю через предоставление качественной desktop-системы. Поиск благоприятных возможностей для ускорения процесса разработки таких решений совершенно естественным образом подвел Mandriva к заинтересованности в покупке Lycoris. Согласно плану, с ус-

ловиями которого согласилась Lycoris, начинается разработка нового продукта, неким образом сводящего Mandriva Discovery и Desktop/LX в одной точке, судя по всему, единого дистрибутива, и эта работа уже началась. По договоренности сторон, основатель Lycoris Джозеф Чик возглавляет работу по интеграции в Mandriva Club уникальных функций Iris Software Gallery и объединению собственных наработок Desktop/LX с технологиями Mandriva. www.mandriva.com

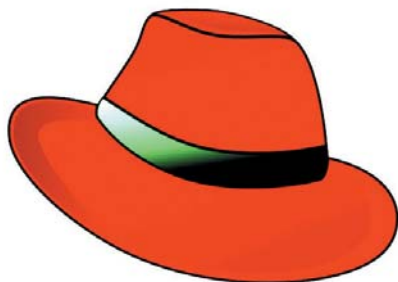
Firefox — продукт года

По версии PC World, продуктом 2005 года назван браузер Mozilla Firefox, который оказался на самой вершине списка 100 лучших продуктов текущего года из мира компьютерной индустрии. В число отмеченных вошли также дистрибутив Ubuntu Linux 5.04, поисковые движки A9 и Google, почтовый клиент Mozilla Thunderbird и онлайн-энциклопедия Wikipedia. PC World особенно выделяет тот факт, что из 100 победителей 23 имеют самую привлекательную цену — они свободны. www.pcworld.com



Вольному — воля!

На прошедшем в Новом Орлеане саммите Red Hat представитель компании Марк Веббинк довел до сведения общественности решение о передаче управления и полного владения проектом Fedora организации Fedora Foundation, создаваемой в настоящее время. Fedora, таким образом, получает полную свободу, а Red Hat демонстрирует очередной пример, достойный украсить страницы любого учебника по менеджменту. Естественно, «молодая поросль» не оставлена на произвол судьбы: Red Hat берет на себя и отеческую заботу, и материнское внимание — спонсорство и участие в работе над Fedora. www.redhat.com



Использование решений, основанных на Linux, продолжает наступление широким фронтом на рынок встраиваемых систем. Производитель армейской тяжелой колесной техники Oshkosh объявил об оснащении шасси Oshkosh A3 HEMTT, системы тактического развертывания, встраиваемой операционной системой Linux, работающей на процессоре ARM 9E в связке с базой данных db.* от ITTIA. Это первый подобный опыт в работе компании, решение в пользу Linux было принято безоговорочно: присущая этой ОС стабильность вкупе с отсутствием нужды в дополнительных лицензионных отчислениях стали главными факторами, повлиявшими на этот выбор.

В ружье!

Проект ITTIA db.* выдержал конкуренцию со стороны других альтернатив благодаря низкой требовательности к системным ресурсам, оптимальной производительности и открытости, что сказалось на конечной стоимости всего решения, позволившего сэкономить сотни тысяч долларов.

<http://linuxdevices.com>



Новый офисный стандарт

OASIS официально объявила о принятии за стандарт Open Document Format for Office Applications версии 1.0, или просто OpenDocument. Пользователям остается попросту дождаться выхода релиза OpenOffice.org 2.0, чтобы познакомиться с новым форматом или воспользоваться свежим KOffice 1.4.

OpenDocument представляет собой свободный, открытый формат, основанный на XML, и имеет неоспоримое удобство в использовании — переносимость и независимость от производителя

программного обеспечения. В Microsoft уже отреагировали на это событие: сначала было объявлено о том, что новые XML-форматы для следующего офисного пакета будут свободными от лицензионных отчислений, а затем поступило и беспрецедентное приглашение к конструктивному диалогу Майкла Тиманна, президента Open Source Initiative, бывшего главного инженера Red Hat, а ныне специалиста по связям с Open Source в Red Hat Inc.

www.oasis-open.org

Qt — начало экспансии?

Компанией Trolltech выпущена новая версия популярной среды разработки кросс-платформенных приложений — Qt 4. Добавлена возможность интеграции с Microsoft Visual Studio .NET, а двойное лицензирование для Windows не оставит в стороне и апологетов Open Source. Президент Trolltech Эйрик Чамбенг заявил: «Qt 4 — самая важная из всех когда-либо выпущенных версий Qt, придающая продукту статус всеобъемлющей структуры для кросс-платформен-

ной разработки. Разработка ПО для нескольких платформ становится очень важной, и Qt позволяет создавать высокопроизводительные приложения, которые работают почти повсеместно. Наша цель с релизом Qt 4 — распространить возможности Qt на большее число приложений: от графических программ для настольных ПК до требовательных к ресурсам приложений и серверного ПО, работающего с базами данных».

www.trolltech.com



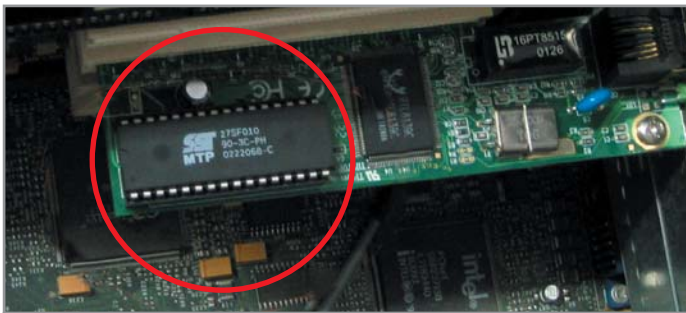
Новая жизнь старых компьютеров

Не спешите разбирать свой компьютер на запчасти. Работать с современным программным обеспечением можно и на стареньком Pentium I, и даже на древней «четверке». Звучит заманчиво? Тогда читайте дальше.

В свое время мне уже приходилось писать небольшую статью о настройке класса бездисковых станций под управлением LTSP (см. в Интернете статью «В бой идут одни «старики»). Сейчас, спустя три года, проект дорос до версии 4.1.1, появились новые возможности и особенности, поэтому есть смысл рассказать о них и заодно поделиться опытом практического внедрения подобной технологии. Так получилось, что в настоящий момент я работаю системным администратором и преподаю в Луганском политехническом колледже (ЛПК) на Украине, поэтому работа терминал-сервера была обкатана на одной из самых «неблагонадежных» и «беспокойных» групп пользователей — студентах. Однако это ни в коей мере не означает, что данный материал не будет представлять интереса для домашних или бизнес-пользователей. В нашем колледже в качестве платформы для терминал-сервера используется ALT Linux Master 2.4.

Преимущества терминал-серверных технологий в Linux

Очевидное достоинство, на которое в первую очередь всегда обращают внимание, обсуждая перспективность данной технологии, — это использование бездисковых станций в качестве терминалов, что снижает аппаратные требования и, как следствие, затраты на приобретение и последующую эксплуатацию оборудования. Несомненно, это так, но, говоря об этом, многие очень часто забывают о дополнительных преимуществах. Во-первых, это централизованное хранение пользовательских данных и обслуживание всего одного экземпляра операционной системы, под управлением которой работает сервер (поддержка в состоянии готовности парка рабочих станций с установленной операционной системой — это кошмар наяву для системного администратора). Следующий плюс заключается в том, что, как правило, бездисковые



В качестве bootrom для сетевой платы RTL-8139C была использована микросхема 27SF010

| Терминал-сервер |

- ▶ Имя: p3.home.lg
- ▶ IP-адрес: 192.168.0.1

| Бездисковая станция |

- ▶ Имя: vm.home.lg
- ▶ IP-адрес: 192.168.0.3

| Установка LTSP |

Установка производится при помощи утилиты `ltsppadmin`, входящей в состав пакета `ltspp-utils` (www.ltspp.org/ltspp-utils-0.11-0.noarch.rpm). `ltsppadmin` — утилита, написанная на основе Perl, которая выполняет закачку и установку пакетов, а также предварительную настройку служб, необходимых для работы ТС. В том случае, если LTSP устанавливается на компьютере впервые (естественно, все действия выполняется с правами администратора), то при выборе первого пункта меню «Install/Update...» вам будет предложено выполнить предварительную настройку: указать источник пакетов, путь для установки дерева каталогов, содержащего файлы, необходимые для работы бездисковой станции, параметры прокси-сервера (опционально) и выбрать группы пакетов.

К вопросу о выборе источника пакетов — вы можете самостоятельно скачать пакеты (<http://ltspp.mirrors.tds.net/pub/ltspp/ltspp-4.1/>) и указать в `ltsppadmin` (см. пункт «Configure the installer options») путь к ним. Либо можете скачать ISO-образ (<http://ltspp.mirrors.tds.net/pub/ltspp/isos/>), содержащий необходимые пакеты и подмонтировать его следующей командой:

```
mount -o loop ltspp-4.1.1-1.iso /mnt/disk
```

В этом случае обратите внимание на формат пути в настройках `ltsppadmin`, он выглядит следующим образом: `file://mnt/disk`.

| Настройка при помощи ltsppcfg |

Для первичной (без ручной доводки никак не обойтись) настройки служб мы можем воспользоваться утилитой `ltsppcfg`, которая запускается отдельно или из `ltsppadmin` (см. пункт «Configure LTSP»). В первую очередь нас интересует пункт «Show the status of all services». Окно разделено на три части. Первая показывает список сетевых интерфейсов. Если их несколько, то необходимо выбрать тот, который будет использовать ТС. Следующая таблица включает список сервисов, необходимых для работы терминал-сервера. Обратите внимание на колонки «Installed», «Enabled» и «Running».

```
ltsppcfg v0.11 The Linux Terminal Server Project (http://www.LTSP.org)

Interface IP Address Netmask Network Broadcast Used
eth0 192.168.0.1 255.255.255.0 192.168.0.0 192.168.0.255

Service Installed Enabled Running Notes
dhcpd Yes no no Version 3
lftpd Yes no no Has '-s' flag
portmapper Yes no no
nfs no no no
xdmcp Yes no no xdm, gdm, kdm Using: kdm

File Configured Notes
/etc/hosts no
/etc/hosts.allow no
/etc/exports no
/opt/ltspp/i386/etc/lts.conf Yes

Configured runlevel: 5 (value of initdefault in /etc/inittab)
Current runlevel: 5 (output of the 'runlevel' command)

Installation dir...: /opt/ltspp

Press <enter> to return to the main menu... █
```

Интерфейс утилиты ltsppcfg прост и понятен

Если вы забыли установить один из пакетов или включить его, таблица сообщит вам об этом. Последняя часть показывает, сконфигурированы ли файлы, необходимые для работы LTSP. В любом случае здесь невозможно что-то изменить, для этого необходимо вернуться на шаг назад и выбрать пункт «Configure the services manually». Рассмотрим подробно каждый из пунктов.

| Runlevel |

Уровень работы системы. Для терминал-сервера необходим пятый уровень.

| Interface selection |

Выбор сетевого интерфейса. На самом деле этот параметр используется при формировании конфигурационного файла `lts.conf`: его значение указывается в ключах `SERVER`, `XSERVER` и т. п. Если в системе присутствует один сетевой интерфейс, то параметр настроен по умолчанию.

| DHCP configuration |

Здесь формируется файл `dhcpd.conf`. Имейте в виду, что `ltsppcfg` формирует файл `dhcpd.conf` в каталоге `/etc`. В то время как в ALT Linux Master 2.4 (и не только в нем) данный конфигурационный файл хранится в `/etc/dhcp`. Поэтому можете создать для него символическую ссылку:

```
ln -s /etc/dhcpd.conf /etc/dhcp/dhcpd.conf
```

Либо можете скопировать созданный файл в этот каталог. Более правильным будет второй вариант — по той простой причине, что данный файл приходится дорабатывать вручную. Ниже приводится пример `dhcpd.conf`, который описывает параметры загрузки для двух бездисковых станций:

```
ddns-update-style none;
default-lease-time 21600;
max-lease-time 21600;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 192.168.0.1;
option domain-name "home.lg";
```

```
option root-path "192.168.0.1:/opt/ltsp/i386";
option option-128 code 128 = string;
option option-129 code 129 = text;
option space PXE;
option PXE.mtftp-ip code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
option PXE.discovery-control code 6 = unsigned integer 8;
option PXE.discovery-mcast-addr code 7 = ip-address;
subnet 192.168.0.0 netmask 255.255.255.0 {
use-host-decl-names on;
option log-servers 192.168.0.1;
host vm {
hardware ethernet 00:0C:29:3F:D2:0F;
fixed-address 192.168.0.3;
if substring (option vendor-class-identifier, 0, 9) = "PXEClient" {
filename "/lts/pxe/pcnet32.zpxe";
option vendor-class-identifier "PXEClient";
vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;
next-server 192.168.0.1;
}
else if substring (option vendor-class-identifier, 0, 9) =
"Etherboot" {
filename "/lts/vmlinuz-2.4.26-ltsp-3";
option vendor-encapsulated-options
3c:09:45:74:68:65:72:62:6f:6f:74:ff;
option vendor-class-identifier "Etherboot";
next-server 192.168.0.1;
}
}
}
```

В нашем случае это сетевые параметры виртуальной машины, созданной в VMware для проверки работоспособности LTSP (MAC-адрес ищите в файле с расширением .vpx, параметр ethernet0.generatedAddress). Виртуальная машина с именем vm использует для загрузки протокол PXE. Обратите внимание на второй параметр filename – в нем указывается имя файла, содержащего образ ядра. В данном случае это ядро 2.4.26. Если хотите, можете изменить его на 2.6.9. Если используется сетевая плата, работающая на ISA-шине, для ее успешной инициализации необходимо указать имя драйвера и номер порта ввода-вывода, например:

```
host ws001 {
hardware ethernet 00:20:FE:00:D3:1C;
fixed-address 192.168.0.100;
filename "/lts/vmlinuz-2.4.26-ltsp-3";
option option-128 e4:45:74:68:00:00;
option option-129 "NIC=ne IO=0x300";
}
```

| TFTP configuration |

Настройка tftp-сервера ограничивается тем, что в файле /etc/xinetd.d/tftp в параметре disable указывается значение

«no». Однако файл требует доработки: дело в том, что в ALM 2.4 tftp-сервер использует в качестве домашнего каталога /var/lib/tftpboot, в то время как инсталлятор LTSP копирует образы ядра в подкаталог /tftpboot. Есть два выхода из подобной ситуации. Первый — отредактировать ключ server_args в файле /etc/xinetd.d/tftp:

```
server_args = -u tftp -s /tftpboot
```

Второй — скопировать в /var/lib/tftpboot содержимое /tftpboot и создать символическую ссылку на него:

```
cp -pvR /tftpboot/* /var/lib/tftpboot
rm -rf /tftpboot
ln -s /var/lib/tftpboot /tftpboot
```

Рекомендуем воспользоваться вторым способом. Кроме этого по умолчанию tftp-сервер будет принимать запросы только с локального интерфейса (см. /etc/xinetd.conf параметр only_from). Для того чтобы снять это ограничение, необходимо в /etc/xinetd.d/tftp добавить строку «only_from += 192.168.0.0/24». В результате /etc/xinetd.d/tftp примет следующий вид:

```
service tftp
{
disable = no
socket_type = dgram
wait = yes
user = root
server = /usr/sbin/in.tftpd
server_args = -u tftp -s /var/lib/tftpboot
only_from += 192.168.0.0/24
}
```

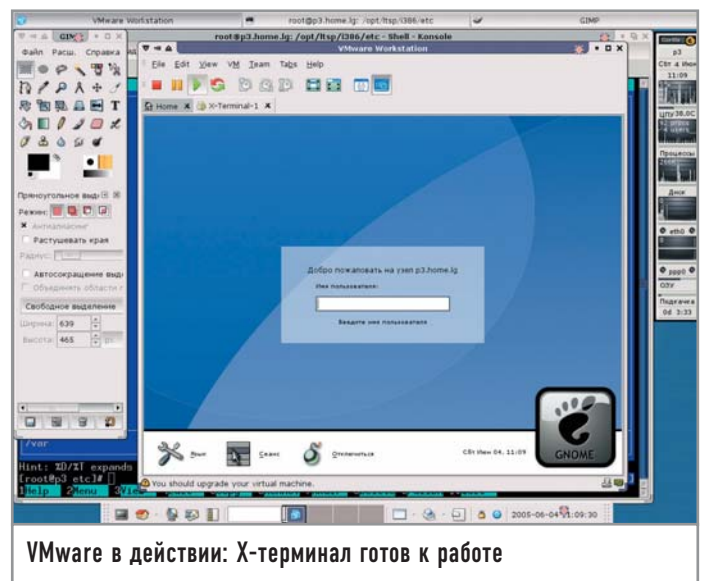
| Portmapper configuration |

Настройка сервиса portmap сводится к его включению. То же самое можно сделать при помощи утилиты chkconfig:

```
chkconfig --add portmap
```

| NFS configuration |

Как и предыдущий, данный пункт предназначен исключительно для включения nfs-сервера.





Компьютер-ветеран «белой» сборки, успешно справляющийся с новой ролью X-терминала

| XDMCP configuration |

В этом пункте вы активируете протокол XDMCP, который используется для удаленного подключения к X-серверу. В зависимости от того, какой именно графический менеджер активен (kdm, gdm и т. п.), изменяется и соответствующий конфигурационный файл (/etc/X11/kdm/kdmrc, /etc/X11/gdm/gdm.conf и т. д.). Обратите внимание, что активный графический менеджер для входа в систему определяется в файле /etc/sysconfig/desktop одним из следующих ключевых слов: KDE, GNOME, WindowMaker, AnotherLevel.

К сожалению, в ALM 2.4 эту операцию придется выполнить вручную. В качестве примера рассмотрим, как это сделать для gdm. Во-первых, припишите в /etc/sysconfig/desktop слово GNOME (естественно пакет gdm должен быть уже установлен в системе). Затем откройте файл /etc/X11/gdm/gdm.conf, найдите в нем секцию [xdmcp] и исправьте опцию «Enable», указав в качестве параметра True (то же самое можно сделать, запустив утилиту gdmconfig). Следующим шагом необходимо разрешить X-серверу «слушать» TCP-запросы. Для этого отредактируйте файл /etc/X11/xinit/xserverrc, убрав ключ -nolisten tcp. После этого перезапустите сервис dm.

| Create /etc/hosts entries |

Этот пункт сгенерирует в файле /etc/hosts 253 записи примерно следующего вида:

```
192.168.0.2 ws002.ltsp ws002
```

Конечно, так много нам не надо, проще прописать вручную необходимые записи. Согласно сетевым параметрам, указанным выше в качестве примера, файл должен выглядеть таким образом:

```
127.0.0.1 localhost.localdomain localhost
192.168.0.1 p3.home.lg p3
192.168.0.3 vm.home.lg vm
```

Либо, если в локальной сети работает DNS-сервер, то можно не указывать имя/IP-адрес бездисковых станций, а прописать их в прямой и обратной зонах DNS. Это необходимо сделать, иначе по NFS не будет монтироваться файловая система.

В колледже для этого используется DNS-сервер, работающий на отдельной машине, и был случай, когда терминал-сервер был включен раньше компьютера, на котором работает DNS-сервер. В результате загрузка бездисковых станций останавливалась на попытке смонтировать через NFS корень файловой системы.

| Create /etc/hosts.allow entries |

Чтобы ограничить доступ к сетевым ресурсам, воспользуйтесь этим пунктом. Он создаст записи следующего вида:

```
bootpd: 0.0.0.0
in.tftpd: 192.168.0.
portmap: 192.168.0.
```

Первая разрешит dhcp-серверу получать пакеты из любого источника. Две другие ограничат прием пакетов для tftp и portmap только из подсети 192.168.0.0/24.

| Create /etc/exports entries |

Этот пункт настраивает nfs-сервер. Он прописывает в /etc/exports каталоги, которые будут доступны через NFS:

```
/opt/ltsp 192.168.0.0/255.255.255.0(ro,no_root_squash,sync)
/var/opt/ltsp/swapfiles
192.168.0.0/255.255.255.0(rw,no_root_squash,async)
```

Первый параметр описывает корень файловой системы, монтируемой через NFS. Второй — каталог, где будут храниться SWAP-файлы. Использование своппинга по сети необходимо в том случае, если объем оперативной памяти на рабочей станции будет меньше 16 Мбайт.

| Create lts.conf file |

Этот пункт создает в /opt/ltsp/i386/etc файл lts.conf, в котором описываются параметры рабочих станций. По умолчанию он содержит следующую информацию:

```
[Default]
SERVER = 192.168.0.1
XSERVER = auto
X_MOUSE_PROTOCOL = "PS/2"
X_MOUSE_DEVICE = "/dev/psaux"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 3
USE_XFS = N
SCREEN_01 = startx
```

Это секция, параметры которой используются по умолчанию. Если параметры отдельных серверов (xfs, telnet и т. п.) в настройках не указаны явно, то они берутся из SERVER. Значение «auto» в XSERVER указывает на то, что X-сервер, запускаемый на рабочей станции, должен автоматически определить видеоплату. Параметр SCREEN_01 указывает, что именно должно быть запущено на первой консоли. В нашем случае это X-сервер. Список того, что можно запустить на стороне бездисковой станции, вы найдете в /opt/ltsp/i386/screen.d. При первоначальной настройке рекоменду-

ется изменить параметр SCREEN_01 так, чтобы иметь возможность запускать текстовую оболочку. Для этого необходимо создать секцию с названием узла и добавить в него следующую строку:

```
[vm]
SCREEN_01 = shell
```

Естественно, что значения, указанные для конкретных машин, будут перекрывать значения по умолчанию.

| Проверка сервисов |

Любая сверхзадача (а настройку терминал-сервера можно смело отнести к такому классу задач) решается путем ее разделения на составные части. В нашем случае для начала необходимо проверить работу отдельных служб и только после этого переходить к удаленной загрузке.

| tftp-сервер |

Проверяем, стартует ли суперсервер xinetd, управляющий доступом к tftp при загрузке системы, и включен ли tftp:

```
# chkconfig --list xinetd
xinetd 0:off 1:off 2:off 3:on 4:on 5:on
```

Отладка LTSP

Использование виртуальной машины

Чтобы облегчить предварительную настройку ТС, рекомендуется использовать эмулятор ПК для платформы x86 VMware (<http://download3.vmware.com/software/wkst/VMware-workstation-5.0.0-13124.i386.rpm>). Продукт коммерческий, но компания-разработчик предоставляет возможность получить пробный ключ на 30 дней (www.vmware.com/vmware-store/newstore/wkst_eval_register.jsp). К приятным неожиданностям следует отнести тот факт, что пользователь имеет возможность получить новый пробный ключ. Правда, при этом ему необходимо в регистрационной форме указать новый электронный адрес. Установка проблем не вызывает, единственный момент, на который необходимо обратить внимание, это изменения в работе конфигурирующего скрипта `vmware-config.pl`, который не может обнаружить переменную среды `ICONDIR` в ALT Linux Master 2.4, содержащую путь к каталогу с иконками. В ре-

зультате скрипт прекращает работу. Есть два способа избавиться от этого. Первый (неправильный) — закомментировать строку номер 7879, содержащую вызов функции `configureDesktopFiles`. Второй (правильный) — добавить в начало файла `/etc/vmware/locations` строку вида: `answer ICONDIR /usr/share/pixmaps`. После этого запуск `vmware-config.pl` проходит без проблем. Из сетевых настроек нас интересует возможность использования виртуального сетевого интерфейса типа `bridged`. В этом случае гостевая операционная система будет «разделять» реально существующую сетевую карту с хост-машиной (компьютер, под управлением которого работает VMware). Стоит учесть, что если на вашем ПК отсутствует сетевая плата, то воспользоваться виртуальным сетевым интерфейсом типа `bridged` не удастся. В этом случае лучше использовать интерфейс типа `host-only`.

6:off

```
# chkconfig --list tftp
```

```
tftp on
```

Затем перезапускаем его:

```
# service xinetd restart
```

```
Stopping xinetd service: [ DONE ]
```

```
Starting xinetd service: [ DONE ]
```

Теперь необходимо проверить работу tftp-сервера. Запускаем tftp-клиент и пробуем получить файл:

```
$ tftp localhost
tftp> verbose
Verbose mode on
tftp> get /lts/vmlinuz-2.4.26-ltsp-3
getting from localhost.localdomain:/lts/vmlinuz-2.4.26-ltsp-3
to vmlinuz-2.4.26-ltsp-3 [netascii]
Received 1975410 bytes in -3.3 seconds [-4752666 bit/s]
```

Если файл не закачивается, нужно проверить, запущен ли tftp-сервер (он «слушает» запросы на 69-м порту):

```
# netstat --inet -pan | grep ":69"
udp 0 0 0.0.0.0:69 0.0.0.0:* 3342/xinetd
```

Если сервис запущен, проверьте правильность указания подсети в опции `only_from`, а также пути в `server_args` и права на файлы, которые вы пытаетесь скачать. Причем отладку рекомендуется проводить при отключенном брандмауэре!

| dhcp-сервер |

Как и в предыдущем случае, проверку работы dhcp-сервера было бы правильно начать с утилиты `chkconfig` и его перезапуска. Затем необходимо убедиться в том, что он «слушает» сетевые запросы:

```
# netstat --inet -pan | grep "dhcpd"
udp 0 0 0.0.0.0:67 0.0.0.0:* 1660/dhcpd
raw 0 0 0.0.0.0:1 0.0.0.0:* 7
1660/dhcpd
```

На этом пока все.

| nfs-сервер и portmap |

Проверьте, стартуют ли сервисы при загрузке системы, и перезапустите их. Чтобы убедиться в работоспособности nfs-сервера, необязательно использовать `netstat` — запустите утилиту `exportfs`, чтобы убедиться в том, что сервер экспортирует файловую систему:

```
# exportfs
/var/opt/ltsp/swapfiles 192.168.0.0/255.255.255.0
/opt/ltsp 192.168.0.0/255.255.255.0
```

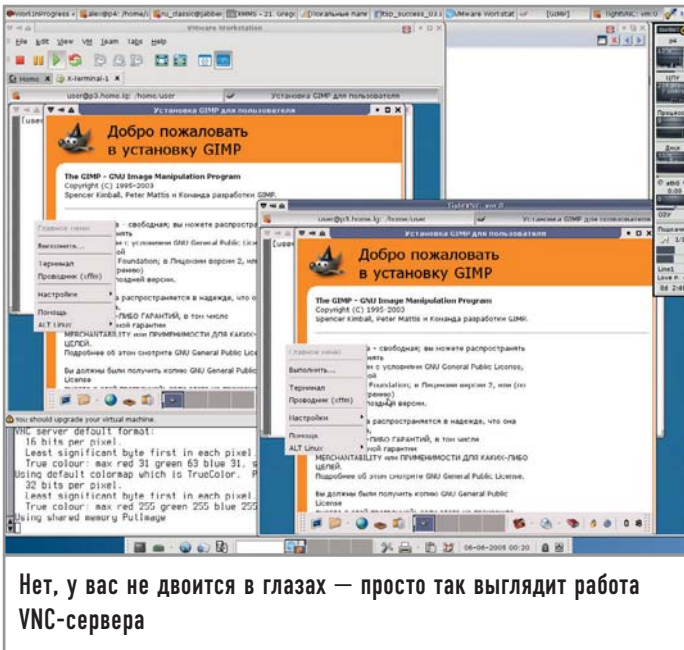
| syslogd |

И еще один важный момент: чтобы иметь возможность получать сообщения о работе системных служб на бездисковых станциях, необходимо на сервере изменить настройки службы `syslogd`, добавив в `/etc/sysconfig/syslogd` ключ `-r`:

```
SYSLOGD_OPTIONS = '-u syslogd -j /var/resolv -r'
```

| Первая проверка |

Итак, пришло время воспользоваться VMware. Сначала необходимо создать виртуальную машину с объемом ОЗУ не менее



32 Мбайт (с большим объемом ОЗУ создавать смысла нет). Эмуляция жесткого диска и CD-привода не обязательна. Главное — не забудьте создать сетевой интерфейс. В BIOS нужно указать загрузку с сетевого интерфейса.

Вернемся к примеру файла `dhcpd.conf`. Обратите внимание на строку `filename "/its/pxe/pcnet32.zpxe";`. Она указывает на файл, содержащий код Etherboot. Фактически протокол PXE выполняет роль поставщика кода Etherboot на ПК. Где можно взять такой код для сетевой платы? На сайте <http://rom-o-matic.net>. Только лучше выбрать версию 5.2.6, так как код из 5.4.0 для сетевой платы, эмулируемой VMware, не будет работать. Вам необходимо выбрать NIC/ROM `pcnet32:lancercsi` и формат `.zpxe`. Полученный файл скопируйте в каталог `/var/lib/tftpboot/pxe` под именем `pcnet32.zpxe`.

Пора включать питание на виртуальной машине. Если вы все сделали верно, то код Etherboot начнет загрузку образа ядра и загрузит shell.

Если загрузка прошла успешно, при запуске «иксов» (`startx`) вы увидите приглашение в систему. Но этого может не происходить по трем основным причинам. Первая — X-сервер в автоматическом режиме не может определить видеоплату. В качестве внешних признаков этого вы увидите мигание экрана и постоянный перезапуск «иксов». В этом случае попробуйте указать в `lts.conf` драйвер. Например, так:

```
XSERVER=s3
```

Причина вторая — X-сервер на стороне ТС «не слушает» запросы. Проверить это можно при помощи `netstat`:

```
netstat --inet -pan | grep 6000
```

Причина третья — вы забыли включить поддержку XDMCP в оконном менеджере.

Кроме того, может быть еще одна причина появления серого экрана — лимит XDMCP-сессий. Такая ошибка появляется, как правило, в процессе эксплуатации, когда количество машин возрастает. По умолчанию в `gdm` этот параметр равен 16. Для его изменения пропишите в секции `[xdmcp]` строку `MaxSessions` и укажите количество сессий.

Переходя от виртуальных машин к реальным

Поговорим об особенностях настройки реально существующих компьютеров. Так как половина компьютерного парка колледжа (не забыли, о чем шла речь во вступлении?) составляют Pentium и Pentium II, то для экспериментов пришлось изъять именно такой ПК для личного использования. Им оказался компьютер фирмы Dell (корпус desktop), процессор Pentium II 233 МГц с пассивным охлаждением, объемом ОЗУ 32 Мбайт, из которых 2 отводятся для интегрированной видеоплаты S3 TrioV64+, и 17-дюймовым монитором. На борту имеется интегрированный звук, а также чип Crystal 4236. Сетевая плата на базе чипсета RTL 8139C, 10/100 Мбит/с. Во-первых, пришлось скачать с сайта <http://rom-o-matic.net> образ для загрузки с дискеты (`.zdisk`) и записать его на флоппи-диск. Во-вторых, описать отдельную секцию в `dhcpd.conf` с правильным MAC-адресом (и не забыть перезапустить `dhcpd`-сервер):

```
host ak {
    hardware ethernet 00:02:44:64:E7:04;
    fixed-address 192.168.0.4;
    filename "/its/vmlinuz-2.4.26-ltsp-3";
}
```

После этого необходимо описать дополнительную секцию в `lts.conf`, где указать параметры, которые будут отличаться от принятых по умолчанию:

```
[ak]
XSERVER = s3
X_MOUSE_PROTOCOL = "IMPS/2"
X_MOUSE_DEVICE = "/dev/psaux"
X_MODE_0 = 1024x768 78.75 1024 1040 1136 1312 768 769
772 800 +hsync +vsync
```

Первый параметр указывает драйвер видеоплаты. Два следующих — `X_MOUSE` и `X_MODE_0` — описывают параметры разрешения экрана, используемые по умолчанию (1024x768, 85 Гц).

В дальнейшем, когда работа терминала будет отлажена, можно будет записать код в `bootrom` (подробная информация на сайте www.remoteboot.ru).

Доступ к локальным устройствам

Наиболее простой способ организации доступа к локальным дискам на терминале — использование Samba-сервера совместно с `suremount` на стороне терминала и поддержка монтирования `smb`-ресурсов на стороне сервера. Для последнего требуется пакет `samba-client`. Что необходимо сделать? Рассмотрим все на примере организации доступа к флоппи-приводу. Во-первых, необходимо скачать и установить исправленную версию скриптов `hotplug` и `gc.localdev` (<http://wiki.ltsp.org/twiki/pub/Ltsp/LocalMedia/localdev-4.1.1-r1.tgz>, см. файл `README`). Во-вторых, в `lts.conf` необходимо указать строку `LOCAL_DEVICE_01 = /dev/fd0:floppy`. Данный параметр описывает имя устройства и его псевдоним. Если теперь перезагрузить терминал, то в `/tmp` (речь идет о каталоге терминала) будет создан подкаталог `/drives`, в котором располагаются точки монтирования для локальных устройств. В нашем случае это `floppy`. При этом на терминале

запущен сервер Samba и доступна одна сетевая папка, /tmp/drives. Уже в таком состоянии можно получить доступ к диску от имени пользователя nobody. Для этого можно, например, запустить Konqueror и указать строку вида smb://vm.home.lg/drives/ (для этого понадобится пакет kde-base-kio-samba) либо смонтировать данный ресурс при помощи утилиты smbmount:

```
smbmount //vm.home.lg/drives ~ /smb -o user=nobody
```

Ресурс <http://wiki.ltsp.org/twiki/bin/view/Ltsp/LocalMedia> предлагает пойти чуть дальше и использовать скрипты, которые бы при авторизации пользователя в системе монтировали ресурсы самостоятельно и размонтировали их при выходе из системы. Скрипты LDA-setup.sh и LDA-shutdown.sh (ссылки найдете на веб-странице) скопируйте в /usr/local/bin. Теперь необходимо настроить графический менеджер входа в систему. Рассмотрим, что необходимо сделать, на примере gdm. Пропишите в /etc/X11/gdm/PostLogin/Default следующие строки:

```
PATH="/usr/bin/X11:/usr/X11R6/bin:/opt/X11R6/bin:$PATH:
/bin:/usr/bin"
/usr/local/bin/LDA-setup.sh
```

Скрипты, располагающиеся в каталоге /etc/X11/gdm/PostLogin/, запускаются после авторизации пользователя в системе. Вызов LDA-shutdown.sh пропишите в /etc/X11/gdm/PostSession/Default. Теперь в домашнем каталоге пользователя появится подкаталог drives, содержащий все подмонтированные ресурсы.

Но есть одна ошибка, которая все портит. Она кроется в supermount. Когда локальные ресурсы монтируются к /tmp/drives, то сначала размер каталогов равен 0 (даже если диск вставлен), и это приводит к тому, что Samba показывает на подмонтированном ресурсе файлы нулевого размера. Но на самом деле это подкаталоги! Совет, который предлагается на <http://wiki.ltsp.org> заключается в следующем: необходимо запускать ls -l /tmp/drives/*. Но он не решает проблему, если дискету вытащить и вставить новую. Выход, который предлагаем мы, более простой: перед тем как обращаться к каталогам, необходимо воспользоваться утилитой touch. Проще всего будет «повесить» на кнопку для перехода к устройствам следующие команды:

```
cd ~/drives
touch *
```

«Прикосновение» к точкам монтирования приведет к тому, что размер каталогов изменится и все заработает. Почти... За исключением поддержки кириллицы в именах файлов. Для этого необходимо модифицировать скрипт rc.localdev, добавив в описание конфигурационного файла Samba строки, указывающие необходимую кодировку:

```
dos charset = CP866
unix charset = KOI8-R
display charset = KOI8-R
```

Заключение

Подведем итоги. Использование ТС на базе Linux — это не научная фантастика или мечта энтузиастов, это реальность. Основной вопрос, связанный с переходом на подобную тех-

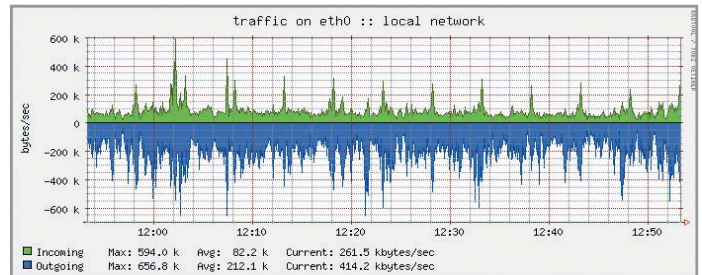


График сетевой загрузки при одновременной работе двенадцати пользователей в среде OpenOffice

нологии, заключается не в том, возможно ли это технически. В первую очередь необходимо решить, а есть ли адекватная замена ПО, которое использовалось до этого. Если рассматривать бездисковые станции как рабочее место, где задействован стандартный офисный набор (офисный пакет, набор программ для работы в Интернете и т. п.), то проблем никаких нет. Но есть программа, которая чаще других упоминается как причина невозможности смены платформы MS Windows на Linux, — 1С. Версии 1С для Linux нет. Но кажется, в этом году намечается прорыв, который позволит решить данную проблему. Компания «Этерсофт» (www.etersoft.ru) разработала патч для WINE (программа, позволяющая запускать Windows-приложения в среде Linux), который дает возможность использовать программу «1С: Предприятие 7.7» (www.etersoft.ru/content/view/59/). Следует отметить, что патч не бесплатный, но в любом случае его стоимость окупается за счет экономии средств при использовании Linux. |

Решение проблем

Раскладка клавиатуры

Проблем при настройке не так уж много. Чаще всего ошибки возникают из-за невнимательности при настройке. Но на одной проблеме остановимся подробно. Дело в том, что разработчики LTSP забыли добавить в проект поддержку нескольких раскладок клавиатуры. Чтобы восполнить этот пробел, необходимо модифицировать файл /opt/ltsp/i386/etc/build_x4_cfg, добавив в секцию (строки 123–129), описывающую настройки клавиатуры, опцию XKBOPTIONS: Option "XkbOptions" "\${XKBOPTIONS:-"grp:ctrl_shift_toggle"}"

То же самое необходимо сделать и в конфигурационном

файле для XFree86-3.3.6 /opt/ltsp/i386/etc/build_x3_cfg (строки 92–97). После этого допишите в lts.conf (либо в секции [Default], либо в секции, описывающей какую-либо конкретную машину) следующие строки:

```
XKBLAYOUT = us,ru(winkeys)
XKBOPTIONS = grp:ctrl_shift_toggle
XKBMODEL = pc105
```

Этот пример описывает конфигурацию для клавиатуры на 105 клавиш (наиболее распространенная модель на рынке) и раскладки на два языка, переключаемые при помощи клавиш «Ctrl+Shift». Для XFree86-3.3.6 XKBLAYOUT принимает просто значение ru.

СИСТЕМНЫЙ ОСМОТР

Раньше достать какой-либо дистрибутив Linux было довольно сложно. На прилавках их не было, продавцы компакт-дисков зачастую вообще не знали, что это такое, а онлайн-магазинов в то время просто не существовало. Поэтому приходилось или закачивать дистрибутив из Интернета или заказывать его «где-то там», а потом и ждать целый месяц или даже больше. Сейчас Linux-дефицита уже нет — заходи в интернет-магазин и выбирай. Только что выбрать?

Действительно, от разнообразия разбегаются глаза. Некоторых начинающих пользователей вводят в заблуждение названия дистрибутивов, например Linux XP. Вот они его и покупают, а потом полностью разочаровываются в Linux. Не скажу, что упомянутый дистрибутив очень плохой, но у неопытного пользователя будет меньше проблем с полной, а не облегченной версией какого-нибудь «серьезного» дистрибутива, например Linux Mandrake или Fedora Core.

В этом небольшом обзоре я попытаюсь облегчить вашу задачу и расскажу о наиболее интересных дистрибутивах — Linux Mandrake 10.1, Fedora Core 3, Slackware 10.1 и Gentoo 2004.3 XOR. Почему именно о них? Так уж исторически сложилось, что изначально было два семейства Linux (разделение связано с разной системой инициализации и произошло оно еще до появления самой ОС — во времена развития Unix). Дистрибутив компании Red Hat — это наиболее яркий представитель первого семейства, а Slackware — второго. Fedora Core — прямой наследник Red Hat Linux, поэтому со Slackware и Fedora Core, я думаю, все ясно. Дистрибутив Linux Mandrake является полностью (ну, скажем, на 99%) совместимым с Red Hat. Компания

Mandrake с самого начала делала упор на начинающих пользователей, и ее дистрибутивы всегда были более дружелюбными, нежели Red Hat: несколько иной набор программ, более удобные средства конфигурирования, больше тем рабочего стола и т. д. Сейчас о степени «дружелюбия» мне судить довольно сложно, поскольку, во-первых, после нескольких лет работы с обоими дистрибутивами большинство операций выполняются на автомате, а во-вторых, в последнее время компания Red Hat создает дистрибутивы, ничем не уступающие, на мой взгляд, по этому критерию продукции Mandrake. В любом случае я постараюсь быть объективным, чтобы у вас не сложилось неправильное мнение о том или ином дистрибутиве.

Еще раз отмечу, что смотреть на дистрибутив в этой статье мы будем глазами начинающего пользователя, поэтому не удивляйтесь, если я не буду приводить рекомендации по исправлению той или иной ошибки: начинающий пользователь не знает, как ее исправить. Во время установки и работы с системой будут проводиться небольшие тесты, а именно: сколько времени занимает процесс установки, загрузка системы, копирование файла и т. д. Время будет измеряться не какой-

нибудь программой, а обычным секундомером, встроенным в мобильный телефон, — это же домашние условия. Все дистрибутивы тестировались на следующей конфигурации ПК: Duron 1,6 ГГц, 256 Мбайт памяти, винчестер 40 Гбайт, видеоплата GeForce FX 128 Мбайт.

Итак, начнем наш обзор с классики — Fedora Core.

| Fedora Core 3 |

| Установка |

Дистрибутив Fedora Core 3 поставляется на четырех компакт-дисках, но программа установки, если не отмечать какие-нибудь дополнительные пакеты, а оставить все как есть, требует только первые три.

Как обычно, загружаемся с компакт-диска, нажимаем «Enter» для начала установки и одновременно запускаем секундомер. Программа установки спросила меня, хочу ли я протестировать установочные диски. Как начинающий и ничего не понимающий в установке Linux пользователь я согласился. Да, в принципе и не хотелось лишиться нормально работающего и уже установленного Linux Mandrake, в случае если что-то не скопируется, ведь FC3 должен быть установлен на его место.

Во время установки я не отходил от компьютера, чтобы оперативно отвечать на запросы программы-инсталлятора. Проверка первого диска не заняла много времени, а от проверки остальных я отказался, нажав кнопку «Continue».

Программа установки очень удобна и проста — даже у начинающего пользователя с ней не будет проблем. Я выбирал «Выборочную установку» и «Ручное разбиение дисков». Неопытным пользователям я настоятельно рекомендую сделать то же самое. Ведь если выбрать другой тип установки, разделы вашего винчестера будут переразбиты, что приведет к потере всех данных. Далее у меня случится небольшая погрешность в измерениях — ведь на моей машине уже есть созданные Linux-разделы, я только изменил тип файловой системы с ext2 на ext3 и выбрал их форматирование. Если у вас до этого не была установлена Linux, понадобится дополнительное время на создание новых разделов и изменение границ уже имеющихся.

Параметры загрузки я выбрал по умолчанию, чтобы не тратить дополнительное время на их установку.

Следующий шаг — выбор программного обеспечения. Я ведь начинающий пользователь, поэтому не знаю, что для чего используется. Просто оставляю все как есть. Как пользователь понимающий, хочу заметить, что FC3 по умолчанию использует среду GNOME, а KDE вообще даже не рекомендуется устанавливать.

Появляется окно «Запуск процесса установки ...», и я перезапускаю секундомер. До этого весь подготовительный этап занял 7 минут и 31 секунду.

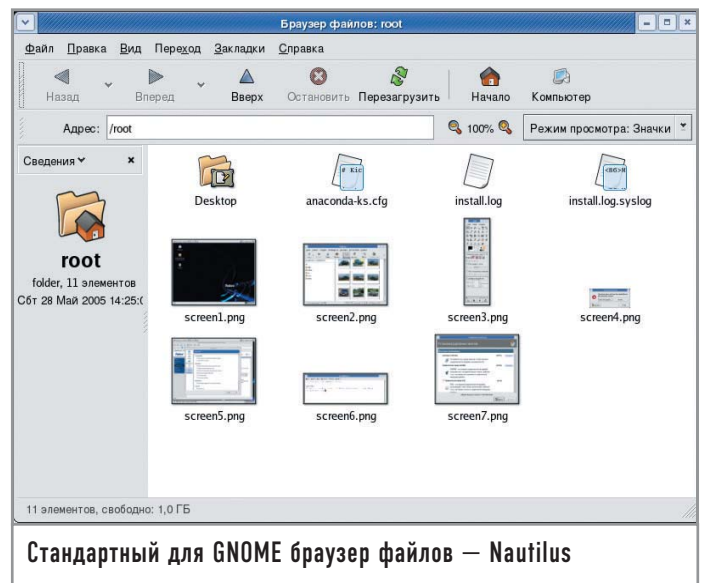
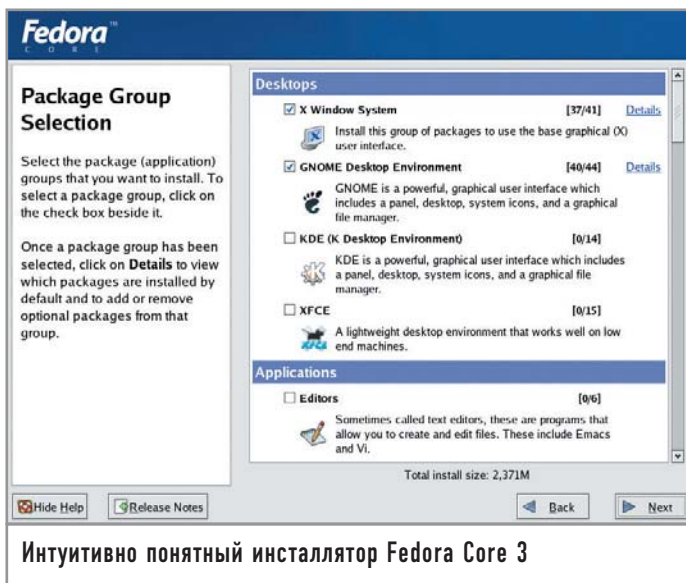
Началось копирование пакетов на жесткий диск. Таймер программы-инсталлятора при копировании бессовестно врал: показывал 15 минут до вставки второго диска, а потом стал показывать 10, хотя прошло уже явно больше пяти минут. Четвертый диск в моем случае (все пакеты — по умолчанию) не понадобился. Установка (до появления окна перезагрузки компьютера) заняла 15 минут и 38 секунд.

После первой перезагрузки система попросила указать некоторые параметры (дату/время, настройки дисплея), добавить новых пользователей, настроить звуковую плату и т. д. Сразу скажу, моя встроенная звуковая плата отказалась работать. Она была опознана, но звука при воспроизведении семпла я так и не услышал. На время оставил все как есть.

| Вторая перезагрузка |

Сразу перезагружаю систему. Включаю секундомер (с момента начала загрузки ядра). Запуск системы (до окна ввода имени пользователя и пароля) занял на моей машине всего 1 минуту и 8 секунд. Запуск GNOME занял еще 7 секунд.

Начинаю работать с системой. GNOME версии 2.8.0 мне очень понравился. Первым делом я попытался выяснить, что же все-таки случилось с моей звуковой платой. Поскольку я начинающий пользователь, на ум мне не пришло ничего кроме проверки уровня громкости. Оказалось, что устройства выключены, поэтому я и не слышал звук! Включаю, устанавливаю максимальную громкость. Запускаю проигрыватель



компакт-дисков и вставляю Extra CD. GNOME предложил мне или открыть диск (чтобы просмотреть дорожку данных), или же воспроизвести (аудиодорожки). Мелочь, а приятно. Выбираю воспроизведение. Все равно звука нет. В общем, звуковую плату я на время оставил в покое и решил посмотреть, что же нового появилось в системе.

Во время исследования я обнаружил, что файловые системы Windows (а у меня четыре Windows-раздела) не подмонтированы. Сделать это вручную особого труда не составляет, но что если человек впервые видит Linux и еле справился с установкой ОС, он ведь не знает, что есть команда mount! Но это уже другой вопрос — как говорится, RTFM перед установкой.

Посмотрим, сколько места осталось после установки системы. Это можно выяснить с помощью программы df. Я устанавливал систему на раздел /dev/hda5, а раздел /dev/hda6 использовал для каталога /home. Сама система (/dev/hda5) заняла 2 446 188 блоков по 1 кбайт — всего 2389 Мбайт. На разделе /dev/hda6 место практически не использовалось, поскольку при установке я не создавал пользователей.

| Программное обеспечение |

По умолчанию устанавливается не все программное обеспечение, а только самое необходимое. Очень понравился удобный просмотрщик изображений Photo Tool.

Также порадовал обновленный GIMP 2.0.5 (хотя я использовал его только для снятия скриншотов), его интерфейс стал более красивым и удобным.

Хочу посмотреть видео, пусть и без звука. Запускаю Helix Player — именно этот проигрыватель используется в FC3 для просмотра видео. Открываю файл, который без проблем воспроизводился в MDK и который был у меня до этого установлен, и вижу следующее: «The player does not have the capabilities to play back this content». Больше комментариев нет.

Далее все как обычно — OpenOffice, Project Planner, диаграммы DIA, утилиты настройки системы. Стоп! Кажется, в группе «Интернет» есть что-то новенькое — Firefox Web Browser. Только почему-то он на английском. Исследовав опции, я так и не нашел выбора языка интерфейса — скорее всего, для его русификации нужно дополнительно установить какой-то пакет. Ну ничего, все равно бороздить просторы Интернета я в нем не буду — скоро установлю другой дистрибутив.

Преимущества	Недостатки
Дружественный интерфейс GNOME 2.8.0	Проблемы с определением звуковой платы
Хороший выбор ПО	Не хватает видекодеков
Удобная программа установки	Проблемы с выключением питания
Новые версии GNOME и KDE	По умолчанию не устанавливаются некоторые привычные программы
	По умолчанию не монтируются файловые системы Windows
	Требуют самостоятельной донастройки после установки
Программное обеспечение	Ядро версии 2.6.9, GNOME 2.8.0, KDE 3.3.0, OpenOffice 1.1.2, GIMP 2.0.5, загрузчик GRUB

| Тесты |

Переходим к самому интересному — тестам. Они будут следующими: тест чтения блока данных с помощью hdparm, тест на время копирования большого файла с раздела FAT32 в раздел ext3 и копирование этого же файла из одного раздела ext3 в другой раздел ext3. Хотел запустить привычный Midnight Commander, но не нашел его и попытался установить с помощью утилиты «Управление пакетами». Она мне не понравилась — отлично подходит лишь для начинающего пользователя, который и сам не знает, что он хочет установить. Мне же лучше было ввести имя пакета и установить его. К сожалению, в этой программе такой возможности не было.

Тратить время на установку МС я не стал, а решил воспользоваться браузером файлов. Может, это и правильно, ведь начинающий пользователь будет использовать именно его, а не какой-то МС, который даже не устанавливается по умолчанию.

У нас есть уже все необходимые программы для проведения тестов. Начнем по порядку — с hdparm. Запускаю «Терминал» (находится в группе «Система» меню приложений). Результат чтения блока размером 174 Мбайт за 3,02 секунды довольно неплохой — это около 57,68 Мбайт/с.

Теперь буду копировать файл размером ровно 700 Мбайт. Это фильм, который мне так и не удалось просмотреть. Получил следующие результаты:

- ▶ копирование с раздела FAT32 (/dev/hda10) на раздел ext3 (/dev/hda6) — 2 мин 32 с;
- ▶ копирование с раздела ext3 (/dev/hda6) на раздел ext3 (/dev/hda5) — 1 мин. 4 с.

Как видите, с «родными» разделами FC3 работает в два раза быстрее. Файловая система ext3 использовалась только потому, что при нормальной установке она предлагается по умолчанию. Ее и выберет начинающий пользователь — он же и не подозревает о наличии других файловых систем.

Все, обзор FC3 на этом заканчивается, и я решил выключить компьютер. При выключении вижу картину из серии «Не ждали»:

Power down

acpi_power_off called

И все. Так машина простояла минут пять, пока я вручную не выключил питание. Сразу вспомнил времена Windows 95: «А теперь питание компьютера можно отключить... вручную».

| Диагноз |

Я так и не понял, для кого разрабатывался этот дистрибутив — для начинающих или продвинутых пользователей? Если для начинающих, о чем свидетельствует дружелюбный интерфейс GNOME, то почему так много недоработок, с которыми этот самый начинающий пользователь не в состоянии справиться. Не работающая звуковая плата (у меня довольно распространенная модель, поэтому такая неисправность будет наблюдаться у многих пользователей), ошибки при воспроизведении видео (очень актуальная задача для домашнего компьютера), неподключенные файловые системы, вдобавок ошибка при выключении питания — и это при том, что я поработал с системой всего лишь час. Поработай я больше, даже не могу предположить, сколько недоработок я бы в ней нашел. Если же дистрибутив разрабаты-

вался для продвинутых пользователей, почему нет привычных им программ (того же MS)? В любом случае я бы не рекомендовал этот дистрибутив начинающим. Сначала нужно поучиться на более легком в освоении продукте, у которого все работает с самого начала, а затем переходить на Fedora Core 3, если, конечно, захочется.

Slackware

Вторым дистрибутивом я решил выбрать Slackware. Просто хотелось на него посмотреть: ведь до этого я видел его в работе только несколько раз. А тут еще и последняя на данный момент версия — 10.1. Дистрибутив поставляется на двух дисках.

Установка

Как обычно, загружаюсь с первого компакт-диска. Вижу экран приветствия:

Welcome to Slackware version 10.1 (Linux kernel 2.4.29)

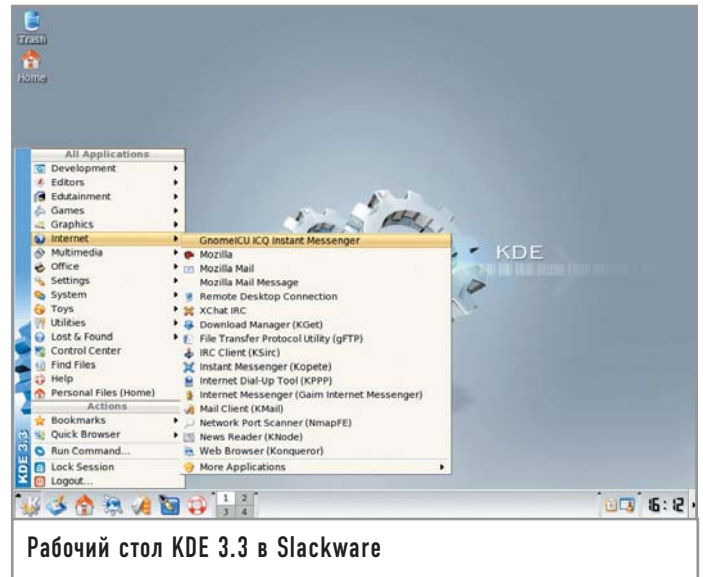
Что? Неужели последний дистрибутив от Slackware основан на старом ядре? Возможно, разработчики посчитали, что новое им не нужно.

Знакомая загрузка ядра предыдущей версии. Программа установки предложила выбрать раскладку клавиатуры, отличную от US:

Enter 1 to select a keyboard map

Неужели установка системы будет происходить в текстовом режиме, да еще и без меню? Нет, меню все-таки появилось, но текстовое. Меню выбора раскладки клавиатуры содержит список файлов раскладок безо всякого описания. Догадайтесь, мол, сами. Выбираю `qwerty/ru.map` и нажимаю «Enter». Честно говоря, текстовая программа установки выглядит как-то архаично. Такое впечатление, что устанавливается одна из первых версий Red Hat. Уже в Red Hat 6.0 программа установки была намного удобнее, хотя тоже работала в текстовом режиме. Затем система предложила ввести пароль пользователя `root`. На данный момент пароля нет, поэтому можно просто нажать «Enter», что я и сделал. Система сообщила, что для изменения таблицы разделов можно ввести команду `fdisk`. Мне повезло — Linux-разделы у меня уже есть, а начинающему пользователю лучше вообще отложить эту затею и попросить кого-нибудь более знающего переразбить жесткий диск. Только на использование текстовой версии `fdisk` может уйти минут пять — это при условии, если знаешь, что делаешь. К тому же `fdisk` не умеет изменять размеры разделов так, чтобы не уничтожить всю информацию. Поэтому лучше использовать для этой цели какую-нибудь другую программу, например Partition Magic для Windows. Если вы все-таки хотите использовать `fdisk`, рекомендую на сайте `dkws.narod.ru` найти статью, в которой я подробно описываю процесс работы с этой программой (http://dkws.narod.ru/linux/many_os.html).

Для начала установки нужно ввести команду `setup`. Появляется меню программы установки. «KEYMAP» я уже выбрал, поэтому отмечаю второй пункт — «ADDSWAP». Он позволяет установить раздел подкачки для системы. У меня он уже создан, поэтому программа запросила разрешения только активизировать этот раздел. После этого появилось текстовое окошко



Рабочий стол KDE 3.3 в Slackware

«FORMAT», в котором нужно выбрать, какие Linux-разделы форматировать, а также файловую систему. По умолчанию предлагается Reiser FS, но для чистоты эксперимента я выбрал `ext3`. Программа установки работает только на английском языке, поэтому, если вы его не знаете, лучше попытаться установить другой дистрибутив.

Затем программа спросила, хочу ли я добавить найденные Windows-разделы в `/etc/fstab`. Соглашаюсь и добавляю два из четырех разделов — для экономии времени. После этого выбираю «Slackware CD or DVD», а также опцию «auto» для автоматического определения диска.

Какие пакеты устанавливать? Отмечаю «full», то есть установку всех пакетов, чтобы не терять время на их выбор. До этого весь подготовительный процесс занял 6 минут и 6 секунд. У вас, скорее всего, этот показатель будет другим. А если у вас еще и нет Linux-разделов, можно смело увеличить это время в два раза.

Время копирования пакетов с двух компакт-дисков заняло 13 минут и 10 секунд. После копирования пакетов программа установки настраивает шрифты, документацию и выполняет другие действия. На все про все ушло 53 секунды.

Вот мы и добрались до самого интересного. Программа установки не знает, откуда брать ядро. Она предлагает несколько вариантов — `bootdisk`, Slackware CD и др. Причем первый — по умолчанию. Я его и выбрал, просто механически так получилось. Программа попросила вставить загрузочную дискету в дисковод `/dev/fd0`. У меня не только нет этой дискеты, но и са-

Преимущества

Дружественный интерфейс KDE 3.3

Высокая производительность файловой системы

Быстрая загрузка системы

Недостатки

Неудобная текстовая англоязычная программа установки

Нет поддержки русского языка файловой системы

Отсутствует пакет OpenOffice

Нет удобного средства разбиения диска — вместо него используется текстовая программа `fdisk`

Программное обеспечение

Ядро версии 2.4.29, GNOME 2.6.2, KDE 3.3, GIMP 2.2.3, загрузчик LILO

мого дисковод. Отказаться — никак. Пришлось, не долго думая, нажать «Reset» и повторить весь процесс заново. Правильный вариант — Slackware CD. Затем выдается список ядер. Я выбрал ядро, которое использовалось при установке:

```
/cdrom/kernel/bare.i/bzImage
```

Следующий этап просто анекдотический. Программа хочет создать загрузочный диск, который она запрашивала на предыдущем этапе. Я отказался от этой возможности. После этого я также отказался и от использования HOTPLUG-устройств. Для экономии времени.

Настройку загрузчика LILO я производил вручную. Не полагаясь на интеллектуальные возможности программы, я выбрал expert и ввел все параметры вручную. LILO установил в MBR. Затем добавил две записи: Linux (/dev/hda5) и Win (/dev/hda1). Обычно эти параметры указываются автоматически — но это в нормальных программах установки... Для инсталляции LILO выбрал команду меню «Install LILO».

После этого нужно указать различные параметры системы — тип мыши, параметры сети (я их не устанавливал), выбор запускаемых сервисов (оставил как есть). Потом программа спросила, хоч ли я добавить свои собственные консольные шрифты. А откуда они у меня? Ясно, что я отказался от этого. Затем — параметры времени (UTC или не UTC, выбор региона, я выбрал Europe/Kiev).

После установки времени нужно выбрать оконный менеджер по умолчанию. Программа предложила KDE, я согласился.

Последний этап — ввод пароля root. Все. Нажимаю заветную комбинацию «Ctrl+Alt+Delete», и система перезагружается. С момента выбора ядра до перезагрузки прошло 4 минуты и 51 секунда.

Загружаюсь. Система встретила меня голой консолью. Ввожу имя пользователя root и свой пароль. Для запуска X Window (и KDE) нужно ввести startx. Некоторые пользователи Linux, даже те, которые уже использовали эту ОС, не подозревают о существовании подобной команды.

| Вторая перезагрузка |

Я перезагружаю систему, чтобы вычислить время второй загрузки. Оно составило 22 секунды до появления приглашения ввести имя пользователя и пароль и еще 22 секунды на запуск X Window и KDE. Итого 44 секунды. После установки на разделе объемом 3 Гбайт осталось 473 Мбайт свободного места.

| Программное обеспечение |

В этом дистрибутиве используется новая версия KDE — 3.3.3, и еще более новая, чем в FC3, версия GIMP — 2.2.3.

KDE русифицировать никак не удалось. Я установил все пакеты, а из доступных языков был только английский. И кому нужен такой дистрибутив?

Существенным недостатком, на мой взгляд, является отсутствие пакета OpenOffice. Вместо него установлен KOffice 1.3.4.

| Тесты |

А вот тут началось самое интересное. Результат hdparm такой же, как и у FC3. 174 Мбайт скопировались за 3,01 с — на 0,01 с быстрее, чем у FC3. При копировании того же самого фильма размером 700 Мбайт я получил следующие результаты:

- ▶ копирование с раздела FAT32 на раздел ext3 — 1 мин. 12 с;
- ▶ копирование с раздела ext3 на раздел ext3 — 31 с.

Вот вам и ядро 2.4. Slackware в два раза быстрее работает с файлами, чем Fedora Core. Файловая система одна и та же — ext3. Уже и не знаю, куда отнести версию ядра 2.4 — к преимуществам или недостаткам. Никуда не буду относить. Пусть это будет особенностью данного дистрибутива.

| Диагноз |

Неудобная текстовая англоязычная программа установки сразу отпугнет начинающих пользователей. Отсутствие OpenOffice и поддержки русского языка явно пошло не на пользу дистрибутиву. Зато это компенсируется скоростью работы файловой системы и более быстрым запуском системы: 44 секунды против 1 минуты и 15 секунд у FC3. Этот дистрибутив можно порекомендовать квалифицированным пользователям, знающим толк в Linux, уж они-то смогут довести его до ума.

| Gentoo |

Данный дистрибутив несколько отличается от всех других, рассмотренных в этом обзоре. Он предназначен для очень подготовленных пользователей, которые хотят собрать дистрибутив «под себя». Новичку здесь делать нечего.

В моем распоряжении был дистрибутив Gentoo 2004.3 XOR (ядро 2.6.9). Загрузка производилась с Gentoo LiveCD. Поскольку на винчестер я этот дистрибутив не устанавливал, опишу только тесты загрузки и файловой системы:

- ▶ загрузка операционной системы (без X Window) — 37 с;
- ▶ копирование файла размером 700 Мбайт с раздела FAT32 на ext3 — 1 мин. 28 с;
- ▶ такой же файл с раздела ext3 на раздел ext3 — 47 с;
- ▶ hdparm: 174 Мбайт было прочитано за 3,03 с; скорость чтения — 57,47 Мбайт/с.

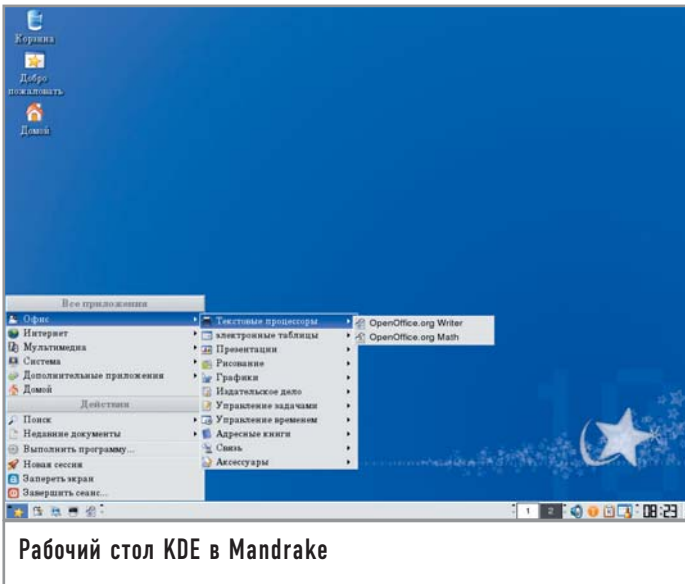
| Linux Mandrake 10.1 |

Этот дистрибутив я специально устанавливал последним: до этого на моем домашнем компьютере был установлен именно он. Как обычно, начнем с инсталляции системы.

| Установка |

Как и Fedora Core 3, Mandrake поставляется на четырех компакт-дисках. Все они нужны при установке — даже если не от-

Преимущества	Недостатки
Простая и быстрая установка	Особых недостатков нет, если не считать немного устаревшие версии GNOME и GIMP
Хороший выбор ПО	
После установки система полностью готова к работе и не требует донастройки	
Нет проблем ни со звуком, ни с видекодеками	
Программное обеспечение	Ядро версии 2.6.3, GNOME 2.4.1, KDE 3.2, OpenOffice 1.1.0, GIMP 1.2.5, загрузчик GRUB



мечать дополнительные пакеты. Программа установки понятна и новичку, с ней практически не бывает проблем — это я знаю точно, потому что устанавливал данный дистрибутив не раз, причем на разные компьютеры.

Весь подготовительный процесс в моем случае (напомню, что я не создавал разделы — они уже были созданы) занял всего 2 минуты и 1 секунду. На копирование пакетов, выбранных по умолчанию, понадобилось 7 минут и 2 секунды. Указание дополнительных параметров (я оставил все по умолчанию, только добавил одного пользователя) заняло 1 минуту и 12 секунд. Итого установка системы заняла 10 минут и 15 секунд.

После перезагрузки приятно обнаружить, что звуковая плата работает, консоль полностью русифицирована, все Windows-разделы подмонтированы (причем подмонтированы правильно — с русскими буквами проблем нет), да и выключается компьютер без всяких осложнений.

| Вторая перезагрузка |

На вторую загрузку (с запуском X Window и автоматическим входом пользователя) понадобилось 42 секунды. После установки на разделе объемом 3 Гбайт осталось свободным 2,1 Гбайт дискового пространства.

| Программное обеспечение |

На четырех компакт-дисках Linux Mandrake вы найдете все необходимое программное обеспечение — от офисных приложений до простых игрушек. Для меня наиболее важны офисный пакет OpenOffice 1.1.0 и графический редактор GIMP 1.2.5. Версия GIMP не очень свежая, но это поправимо — всегда можно загрузить новую.

В качестве оконной среды по умолчанию используется KDE 3.2 — довольно удачная и быстрая версия.

С воспроизведением видеофайлов нет никаких проблем — все фильмы, имеющиеся у меня на Windows-разделах, были воспроизведены проигрывателем Totem — даже те, для которых я в Windows выкачивал из Интернета дополнительные кодеки.

| Тесты |

Дополнительно описывать тесты я не буду, приведу уже готовые результаты:

- ▶ Hdparm: 174 Мбайт прочитано за 3,02 с; скорость чтения — 57,55 Мбайт/с;
- ▶ копирование файла размером 700 Мбайт с раздела FAT32 на раздел ext3 — 1 мин. 26 с;
- ▶ копирование файла размером 700 Мбайт с раздела ext3 на раздел ext3 — 36 с.

| Диагноз |

Linux Mandrake — отличный дистрибутив для начинающих. После установки пользователь получает «готовую к употреблению» систему. Все, что ему нужно настроить, — это сеть и подключение к Интернету. Никаких особых проблем ни при установке, ни при работе с дистрибутивом не возникло.

| Подытожим... |

В этой статье мы рассмотрели четыре дистрибутива: Fedora Core 3, Slackware 10, Gentoo 2004.3 XOR и Linux Mandrake 10.1.

Оптимальным выбором для начинающего Linux-пользователя становится Linux Mandrake. На втором месте — Fedora Core 3. Если же первым дистрибутивом начинающего линуксоида станет Slackware, то, боюсь, что это будет его последний дистрибутив: он отпугнет даже самых смелых. Если Fedora Core 3 и требует дополнительной настройки, то во всяком случае не такой, какая нужна Slackware. Последний больше подойдет для Unix-пользователя, который всю жизнь проработал с BSD-системами, а теперь хочет перейти на Linux (впрочем, такие пользователи — большая редкость). Gentoo — это выбор профессионалов, новичок просто не поймет, что там нужно делать, и у него вполне резонно возникнет вопрос: «А зачем, собственно, я купил этот дистрибутив?»

Что касается производительности, то самым нерасторопным оказался дистрибутив FC3. Даже не могу понять, почему, неужели из-за GNOME 2.8? Ведь результат копирования файла, показанный в Gentoo, приблизительно такой же, как и результат MDK. А версия ядра у Gentoo и FC3 одна и та же. MDK показал второй результат после Slackware. Возможно, сказался необычный подход разработчиков Slackware — предыдущая версия ядра и последние версии программного обеспечения (KDE 3.3). Gentoo в плане производительности показал тоже очень хорошие результаты — ведь загрузка самого дистрибутива производилась с LiveCD. |

	Fedora Core 3	Linux Mandrake	Slackware	Gentoo
Ядро	2.6.9	2.6.3	2.4.29	2.6.9
Общее время установки (мин.:с)	23:09	10:15	20:09	— (LiveCD)
Общее время загрузки	1:15	0:42	0:44	0:37 (LiveCD)
Скорость чтения, Мбайт/с	57,68	57,55	57,68	57,47
FAT32 → ext3	2:32	1:26	1:12	1:28
Ext3 → ext3	1:04	0:36	0:31	0:47

Домашняя видеостудия

Мы не будем касаться темы профессиональной работы с видео и апеллировать к продуктам вроде Avid, сравнивая их с программами для Linux. У нас более приземленные цели — осветить спектр софта, предназначенного для сугубо бытовой работы с видео в Linux.

Разумеется, это не означает, что никто из мира профессионального видео не использует Linux: есть люди, которые создают в Linux видеоролики, а уж о том, что голливудские студии зачастую делают постобработку картин или рендеринг 3D-графики именно в Linux, писали уже многие IT-издания. Взять хотя бы новый мультфильм «Мадагаскар».

Для начала определим темы, которые собираемся рассмотреть. Во-первых, это захват видео с внешнего источника. Во-вторых, конвертирование видео из одного формата в другой. В-третьих, нарезка видеоматериала. И в-четвертых — монтаж видео.

Эта статья окажется полезной, если вы:

- ▶ собираетесь конвертировать свои видеоматериалы с кассет в цифровой формат, а также принимать и записывать телевизионный сигнал через карту тюнера;
- ▶ имеете некоторое количество отснятого домашнего видео (пусть даже сделанного обычным цифровым фотоаппаратом) и хотите хорошо смонтировать его для показа друзьям;
- ▶ желаете конвертировать видео из одного формата в другой (например, из DVD в MPEG-4) и подыскиваете для этого оптимальный набор программ;
- ▶ занимаетесь на дому созданием коммерческих видеороликов;
- ▶ хотите просто иметь представление о возможностях Linux как платформы для обработки видео.

Знание — отличная штука, с которой невозможно попасть впросак. Не так давно на одном уважаемом сайте была опубликована большая статья, где автор задавался вопросом: какая

же система лучше для работы с видео — Windows или Linux? И сам себе отвечал: конечно же, без сомнения, Windows! Хотя он признался, что специально вопрос не изучал, только слышал об одной программе захвата видео под Linux и милостиво предположил о существовании еще нескольких.

Мы не будем говорить, какая платформа лучше, а какая хуже, поскольку какие-то операции удобно делать в одной системе, а какие-то — в другой.

Многие пользователи, имеющие дело с обработкой видео, предпочитают платформу Windows и примерно следующий набор софта: bTv (прием/запись ТВ-сигнала), VirtualDub либо NunDub (захват видео, первичная обработка, конвертирование) и Adobe Premiere либо Pinnacle (монтаж). Из этого комплекта бесплатны разве что Virtual Dub и NunDub. А если добавить к этому разные дополнительные программы-конвертеры и кодеки, цена программного обеспечения перевалит за несколько тысяч долларов.

В мире Linux уже сложилась традиция некоммерческого ПО, поэтому платить вам придется разве что за Интернет, если нужной программы или библиотеки не окажется в вашем дистрибутиве Linux. Среди других преимуществ Linux — стабильность самой системы и возможности файловых систем (можно использовать очень быструю ReiserFS). Ссылки на программы мы будем давать в тех случаях, если утилиты пользуются не столь широким спросом и их нет в стандартном наборе ведущих дистрибутивов Linux.

Захват видео

В современных дистрибутивах получение данных с цифровой камеры или фотоаппарата доведено до полного автоматизма. Например, в Mandriva и Fedora/Red Hat используется программная связка HAL + Gnome volume manager. Подключаете видеокамеру или фотоаппарат к USB-порту, и в зависимости от специфики подсоединенного устройства система выполняет действие. Она может предложить вам импортировать с камеры фотографии и клипы, автоматически запустив программу FLPhoto или GTKam. К примеру, дешевый цифровой фотоаппарат от Mustek просто монтируется к файловой системе, и содержимое его памяти доступно из каталога /mnt/removable1. Для импорта видео с камер, подключаемых через интерфейс FireWire, можно использовать ставшую почти стандартной утилиту Kino. Кстати, Kino позволит выполнять и некоторые операции линейного видеомонтажа.

Если же надо записать видео с видеомagneфона либо телевизионного канала, то одной из самых удобных программ для этого является XdTV (<http://xawdecode.sourceforge.net>).

Для нормальной работы XdTV у вас в компьютере должна быть установлена карта ТВ-тюнера, которая поддерживается Linux. XdTV может принимать сигнал как с обычной ТВ-антенны (или кабельного ТВ), так и с подключенного через композитный разъем устройства, либо через гнездо S-Video. Кроме того, в XdTV встроен программный приемник телетекста — AleTV.

К числу полезных функций XdTV можно отнести мозаичный браузер каналов (в отдельное окно выводятся уменьшенные кадры с каждого канала для наглядности переключения между ними), редактор каналов с возможностью назначения каналам горячих клавиш, секция настройки графических параметров (яркость, контрастность и т. д.), функции сглаживания гребенки, запись определенного канала в заданное время.

Для Linux существует достаточно большое количество таких программ-тюнеров, однако не все они, в отличие от XdTV, умеют записывать видео в файлы. XdTV поддерживает несколько режимов: несжатое видео; FFMPEG — MPEG-4

(по сути, аналог DivX 5), FFMPEG — Mpeg1Video (увы, при выборе этого режима XdTV аварийно завершала работу), FFMPEG — Huffvuv (сжатие без потерь) и XviD (еще один MPEG-4-совместимый кодек).

Частое употребление слова FFMPEG в названиях режимов указывает на то, что для сжатия в упомянутые форматы используется одноименный кодек, который надо установить дополнительно (благо это не сложнее установки любой другой программы в Linux). FFMPEG предоставляет не один, а целый набор кодеков — этим и объясняется его вездесущность.

Внимание! До начала сборки FFMPEG из исходника надо включить опцию «enabled-shared», чтобы создать разделяемые библиотеки. Вот все необходимые для этого команды:

```
./configure --enabled-shared
make
make install
ldconfig
```

Последняя команда нужна для того, чтобы на всякий случай обновить кеш библиотек.

Если у вас большой винчестер и важно качество изображения, наилучшим вариантом будет выбор захвата видео вообще без сжатия, а уже потом вы сможете обработать полученный файл какой-нибудь утилитой вроде Mencoder (о ней речь пойдет ниже).

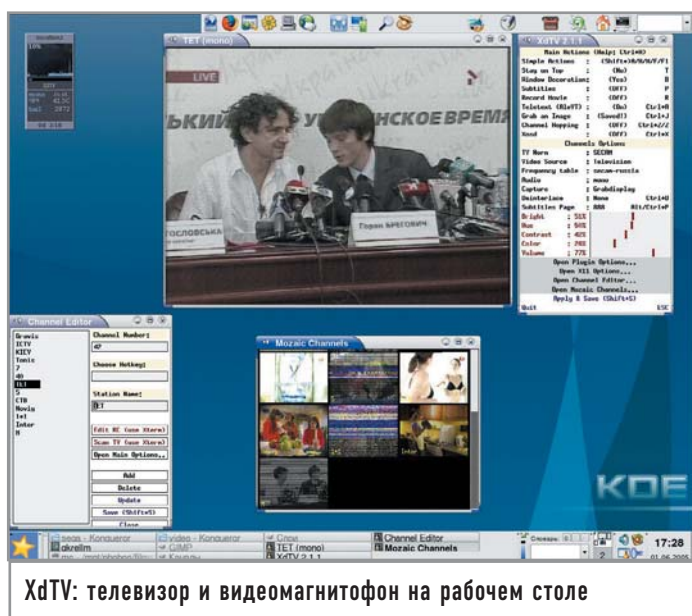
ТВ-программы можно записывать прямо в MPEG-4, если ресурсы вашего компьютера позволяют осуществлять сжатие в реальном времени. Заметим, что включенная функция сглаживания при определенной конфигурации может пожирать довольно много ресурсов процессора. Если компьютер слабый, можно включить запись, а само окно XdTV свернуть — расход вычислительных ресурсов при этом значительно снижается. А при записи с видеомagneфона имеет смысл установить разрешение побольше и оцифровать без сжатия (при наличии огромного жесткого диска) или в режиме «Huffvuv», а уже после обрабатывать и сжимать с помощью какой-нибудь другой программы. Что касается битрейта видео в MPEG-4, то для ТВ подойдет от 900 до 1000, для сигнала с видеомagneфона можно поставить выше.

Звук в XdTV записывается с MP3-сжатием либо без него. Для этого используется кодек LAME (<http://sourceforge.net/projects/lame/>), который, как и в случае с FFMPEG, устанавливается отдельно. Настраивая кодек в соответствующем окне XdTV, лучше не включать переменный битрейт (VBR — Variable). Оптимальным решением будет выбор CBR (Constant) — постоянного битрейта. В случае использования переменного могут возникнуть проблемы синхронизации видео и аудио при последующем редактировании записанного файла (например, если вы захотите вырезать из фильма рекламу).

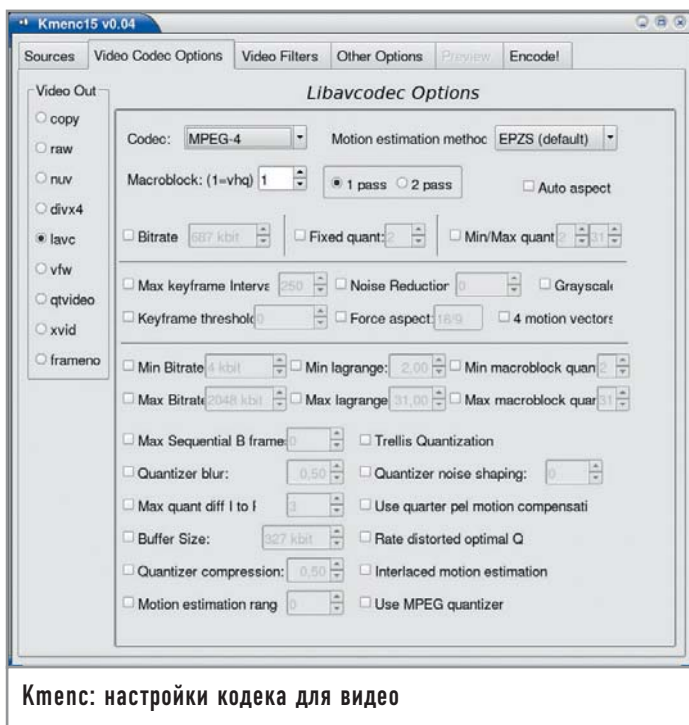
В заключение стоит отметить, что программу XdTV лучше (и проще) установить из исходника. Проще потому, что наверняка можно удовлетворить ее зависимости от разных библиотек. А лучше по той причине, что на этапе конфигурации исходника будет включен флаг оптимизации под ваш процессор.

Обработка

Имея на руках оцифрованный видеоматериал, можем приступить к его обработке. Собственно, этому и будет посвящена ос-



XdTV: телевизор и видеомagneфон на рабочем столе



Кмепс: настройки кодера для видео

тавшаяся часть статьи. Обработку условно разделим на базовую (конвертирование в другие форматы, нарезка и так далее) и продвинутую (видеомонтаж).

Пожалуй, нельзя найти лучшей утилиты для конвертирования, чем Mencoder — родного брата популярнейшего плеера Mplayer, поставляемого с ним в одном архиве. Mplayer и Mencoder входят в состав практически любого популярного дистрибутива Linux. Mencoder — чисто консольная утилита, поэтому любителям графического интерфейса придется обзавестись таковым от сторонних разработчиков. Например, очень хороши в этом плане KMenc15 (<http://kmenc15.sourceforge.net>) и AcidRip (<http://untrepid.com/acidrip/>). Однако имеет смысл освоить всю эту магию командной строки, изучить сотни параметров Mencoder и ощутить себя действительно всемогущим. Ведь только посредством командной строки можно выжать из Mencoder максимум возможностей и сделать с видео именно то, что вам нужно. Кроме того, графические интерфейсы к Mencoder зачастую не поспевают за развитием своего консольного симбионита и не предоставляют интерфейс к его новым настройкам.

В отличие от Windows-программ VirtualDub и NunDub, Mencoder не нуждается во внешних кодеках, а таскает кодеки вместе с собой, на пару с Mplayer. Список форматов, с которыми умеет работать Mencoder, столь внушителен, что можно с уверенностью утверждать: нет такого используемого в настоящее время формата видео, который бы не распознавался с помощью Mencoder (за исключением некоторых особо закрытых). Форматов же, в которые он может конвертировать видео, несколько меньше, но все равно достаточно. Их список можно получить, задав в консоли команду:

```
mencoder -ovc help
```

В зависимости от имеющейся конфигурации Mencoder вы получите список кодеков. У нас, например, на консоль выводится такой результат:

Available codecs:

copy — frame copy, without re-encoding. Doesn't work with filters.
frameno — special audio-only file for 3-pass encoding, see DOCS.
raw — uncompressed video. Use fourcc option to set format explicitly.

nuv — nuppel video

lavc — libavcodec codecs — best quality!

vfw — Vfw DLLs, currently only AVID is supported.

qtvideo — QuickTime DLLs, currently only SVQ1/3 are supported.

xvid — XviD encoding

Обратите внимание на кодек copy. Используя его, вы скопируете в новый файл видео в том же формате, что и оригинал. При кодировании с помощью Mencoder надо выбирать кодек для видео и кодек для звука. Формат командной строки упрощенно таков:

```
mencoder -ovc кодек_видео -oac кодек_звука исходный_файл  
-o файл_результата
```

Можно выбрать подходящие кодеки, причем как для видео, так и звука, и задать их параметры. Для сжатия в MPEG-4-совместимое видео мы рекомендуем использовать LAVC — тот же FFmpeg, набор доброй дюжины кодеков, в том числе MPEG-4, MJPEG, WMV версий 1 и 2, MPEG-1, MPEG-2 и Huffuuv.

Вот простой пример, в котором мы конвертируем файл out.avi (предположим, что он сжат в MPEG-4) кодеком MPEG-2 и выводим результат в файл out.mpg:

```
mencoder out.avi -oac copy -ovc lavc -lavcopts  
vcodec=mpeg2video -o out.mpg
```

Как видите, сначала мы пишем, что видеокодером у нас будет служить LAVC: `-ovc lavc`. Затем передаем LAVC опции, указывающие на использование видеокодера `mpeg2video`:

```
-lavcopts vcodec=mpeg2video
```

Приведем более сложный пример, в котором фильм с DVD конвертируется в MPEG-4:

```
mencoder dvd://2 -oac mp3lame -ovc lavc -lavcopts  
vcodec=mpeg4:vbitrate=1000:vqmin=2:vqmax=25:vhq:v4mv:trell:  
autoaspect -vf scale=512:384 -lameopts cbr:br=128:aq=0  
-o my_movie.avi
```

Как уже говорилось, имеет смысл почитать документацию к Mencoder — правда, русская версия руководства переведена лишь частично, и раздел о кодировании видео как раз не переведен. Поэтому учите английский.

На самом деле в приведенном выше примере ничего сложного нет. Мы выбираем в качестве аудиокодера mp3lame, а для видео — опять-таки lavc. Для последнего задаем опции параметром `-lavcopts`: встроенный кодек — MPEG-4, битрейт видео — 1000, минимальные и максимальные значения квантаizersа — 2 и 25 соответственно. Важен параметр `vhq` — он указывает на то, что сжатие производится с высоким качеством, не в реальном времени, что позволяет избежать всевозможных артефактов вроде выпадения кадров. Параметром `-vf scale=512:384` мы применяем фильтр для видео, задавая фильму другое разрешение — 512:384. Аналогично для LAME мы тоже передаем опции параметром `-lameopts` — это `cbr:br=128` (постоянный битрейт 128) и `aq=0` (наилучшее качество обработки).

Итак, при использовании Mencoder помимо кодеков можно применять также фильтры, среди которых масштабирование и изменение размеров, размывание, перевертывание, отзеркаливание, улучшение резкости и многое другое. Другой пример — конвертирование видео CD в MPEG-4. Это совсем просто:

```
mencoder vcd:// -ovc lavc -lavcopts vcodec=mpeg4 -oac copy
-o out.avi
```

Источник видео читается с устройства vcd, звук копируется в оригинальном формате, для видео ставим кодек lavc/mpeg4. В результате получаем файл out.avi.

Еще одна интересная возможность Mencoder — разбиение большого файла на два. Проще показать это на примере. У нас есть файл movie.avi объемом 1,5 Гбайт. Мы хотим разделить его на два AVI-файла, чтобы записать каждый на болванку. Эти части мы поместим в файлы part1.avi и part2.avi. Оригинал будет разрезан в месте, заданном временем 1 час 13 минут 14 секунд.

Вначале получим первую часть — part1.avi:

```
mencoder -ovc copy -oac copy -endpos 1:13:14 -o part1.avi
movie.avi
```

А теперь вторую — part2.avi:

```
mencoder -ovc copy -oac copy -ss 1:13:14 -o part2.avi movie.avi
```

Как видите, мы ничего не конвертируем — просто используем кодекы copy для звука и видео. Но в первом случае, для part1, мы установили временной предел, до которого копируется фильм: -endpos 1:13:14. Затем, для вывода в файл part2, мы начинаем копировать фильм с временной позиции 1:13:14 — задается это параметром -ss 1:13:14.

Однако не надо сбрасывать со счетов и Mplayer. Его тоже можно использовать для конвертирования видео в последовательность отдельных файлов-изображений, по одному файлу на кадр. Вот как можно перегнать фильм movie.avi в огромное количество пронумерованных PNG-файлов:

```
mplayer -vo png movie.avi
```

Аналогично можем вывести фильм в форматы TGA и JPEG:

```
mplayer -vo tga movie.avi
```

```
mplayer -vo jpg movie.avi
```

Что до звука, то Mencoder может кодировать его в нескольких форматах (кстати, полный их список можно получить, набрав в консоли команду mencoder -oac help). Самый простой из них — обычное копирование исходного звукового потока в новый

файл. Этот режим называется copy. Далее идет PCM — аналог WAV-файла, то есть звук без компрессии. При кодировании в MP3LAME используется установленный отдельно кодек LAME. И, наконец, LAVC — универсальный FFMPEG, который умеет работать и со звуком. В частности, FFMPEG поддерживает AC3, IMA Adaptive PCM, MPEG Layer 2 и MPEG Layer 3. Одним словом, LAME не совсем обязателен, поскольку есть FFMPEG.

Работа с Avidemux 2

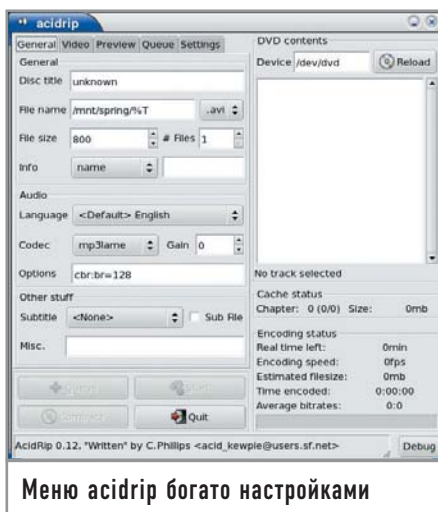
Рассмотрим еще одну утилиту — Avidemux 2 (<http://avidemux.berlios.de>). Ее можно соотнести с такой программой для Windows как VirtualDub — правда, она не столь зависима от установленных в системе кодеков/библиотек, хотя и может использовать внешние библиотеки LAME, MAD, XviD, OggVorbis и другие. Сама же Avidemux 2 поставляется с незаменимым Lavc/FFMPEG, поэтому даже без помощи извне может выполнять полезные задачи.

Например, в Avidemux 2 удобно вырезать рекламу из записанных с телевизора фильмов. В программе есть два маркера, обозначенные на панели инструментов буквами A и B. Перематываете фильм к месту, где начинается реклама, и ставите там маркер A. Затем отматываете к концу рекламы и обозначаете это место отметкой B, нажимаете кнопку «Delete» — и так с каждой рекламой. Впоследствии нужно сохранить результат в новый файл.

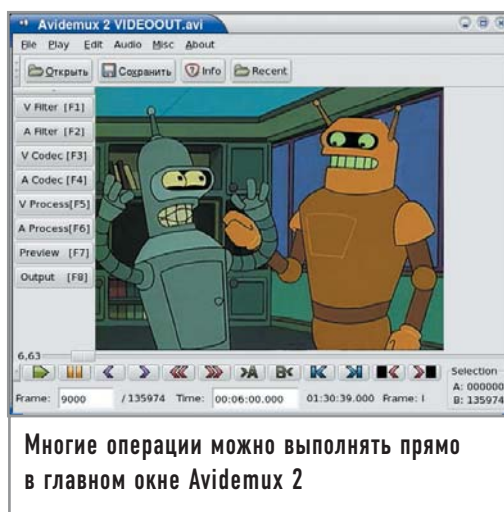
Общий метод работы с Avidemux 2 таков: вы загружаете видеофайл, редактируете его, выбираете (если нужно) другие кодекы для сжатия видео и звука, а затем сохраняете. В процессе сохранения происходит кодирование, если таковое выбрано.

Кодеки для видео и звука выбираются нажатием на кнопки «V Codec» и «A Codec» соответственно. Но чтобы избранные кодекы применились, не забудьте включить кнопки «V Process» и «A Process», при этом, если вы не хотите использовать другой кодек для звука или видео, просто не включайте.

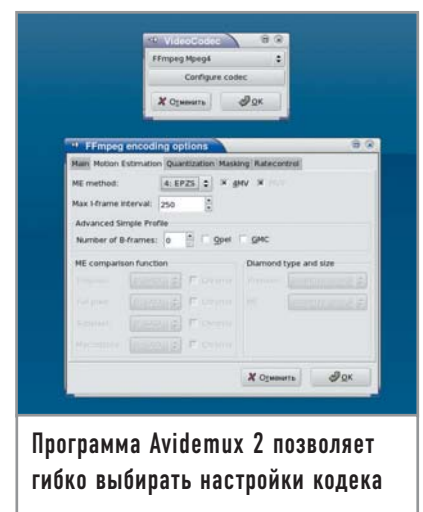
Эти же кнопки влияют и на применение к результирующему файлу различных эффектов. Эффекты и их настройки скрываются под кнопками «V Filter» и «A Filter». Всего видеопределителей несколько десятков, среди которых, кроме подавления шумов, наведения резкости, сглаживания и изменения размеров, есть и более нестандартные вещи, такие как



Меню acidrip богато настройками



Многие операции можно выполнять прямо в главном окне Avidemux 2



Программа Avidemux 2 позволяет гибко выбирать настройки кодека

Subtitle, который позволяет отрендерить субтитры прямо в фильм. Выбирается внешний файл с субтитрами, а также его кодировка, шрифт субтитров, размер и цвет.

Этим функциональность Avidemux 2 не ограничивается. Хотите экспортировать во внешний файл звуковую дорожку фильма? Выделяете нужный фрагмент маркерами (или не выделяете, если нужен звук из всего фильма) и применяете пункт меню «Audio → Save audio».

Нужно заменить звуковую дорожку? Проще простого — подгружаете внешний звуковой файл с ней (от WAV до MP3 и AC3) из меню «Audio» и сохраняете фильм. В итоге старая звуковая дорожка будет заменена на новую, из файла. Можно добавить к фильму и вторую звуковую дорожку — через меню «Audio → Second audio track».

Замечание о звуке. Если вы редактируете фильм, сжатый в MPEG-4, а звуковая дорожка к нему имеет переменный битрейт, то для звука придется составить временную карту. Делается это с помощью пункта меню «Audio → Build VBR time map».

Итак, подведем итог работе с Avidemux 2. Это очень удобная программа для редактирования видео с хорошо подобранным набором встроенных инструментов и функций. Кроме того, в ней можно осуществлять базовый монтаж. Но могут возникнуть некоторые трудности с установкой из исходника. Например, в дистрибутиве Mandriva 2005 LE удалось собрать (и то после ручной правки конфигурационных файлов) Avidemux 2 версии 2.0.26, хотя есть и более свежий релиз.

Тематически мы подошли к завершающей части обработки видео, а именно нелинейному монтажу.

Монтаж

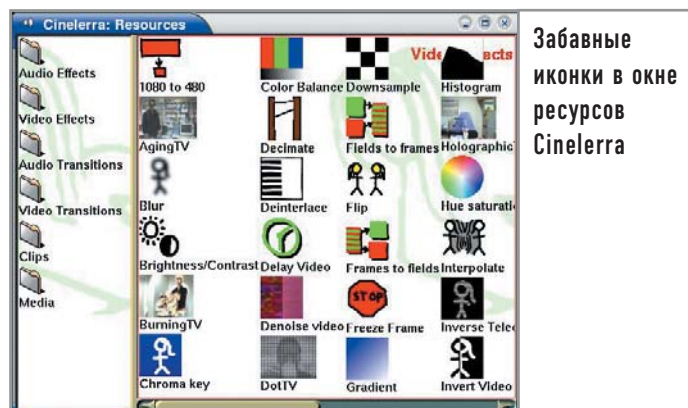
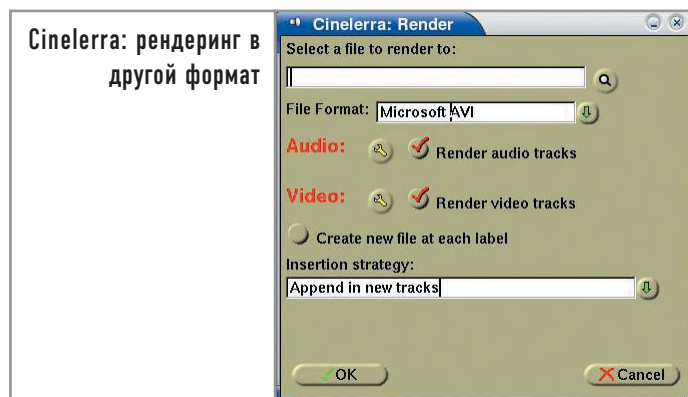
Слово-то какое важное — монтаж! Сразу мысленно представляются доменные печи, заводские громадины корпусов... Программы для нелинейного монтажа видео обычно имеют такой же колоссальный размах.

Сама по себе эта область достаточно специфична, не востребована массами, и в близких к анархии условиях Linux пока не получила должного распространения. Коммерческие продукты для видеомонтажа выпускаются в основном для Mac и Windows и имеют заоблачные цены (впрочем, студии могут себе такое позволить). Однако есть альтернатива.

Если нужна простая в использовании утилита для линейного монтажа вроде Windows Movie Maker, то в Linux вам подойдет продукт, носящий многозначительное имя Kino. Для более серьезного дела, нелинейного монтажа, понадобится Cineerra (<http://heroinewarrior.com/cineerra.php3>).

Работа с Cineerra

Создание Cineerra окутано тайной. Над этим продуктом трудится группа разработчиков, называющая себя Heroine Virtual Ltd. Все они работают в крупных компаниях, руководству которых не нравится, когда сотрудники делают что-то на стороне. Поэтому программисты Cineerra вынуждены скрываться от публики за вывеской виртуальной компании Heroine Virtual Ltd. Помимо Cineerra они ведут еще несколько проектов, относящихся к мультимедиа: плеер XMovie, библиотеки Libmpeg3 и



QuickTime 4 Linux, микшерные утилиты Mix2000 и Mix2005. Установка Cineerra может потребовать некоторых усилий. Есть два варианта: собирать продукт из исходника либо устанавливать пакет. После установки PRM-пакета (командой `rpm -i -f -force --nodeps` пришлось сделать символическую ссылку на одну библиотеку, которая в Mandriva называлась иначе, нежели в Fedora, для которой и был собран пакет. Вообще, предпочтительнее исходники, однако Cineerra — продукт большой, и архив с исходниками больше пакета в несколько раз.

Требования к аппаратуре у Cineerra довольно высокие (но меньше, чем у Windows-версий продуктов Avid). Для станции по обработке видео это: двойной Opteron 2,4 ГГц с 4 Гбайт памяти, а для узла кластера рендеринга (Cineerra поддерживает распределенный рендеринг по сети) — одинарный Athlon 2,4 ГГц и 512 Мбайт памяти. Впрочем, на одном мощном процессоре Cineerra работает тоже хорошо без всяких кластеров.

Между прочим, как вы заметили, в отличие от Windows-продуктов для обработки звука, Cineerra (как и автор этой статьи) отдает предпочтение процессорам AMD. Более того, разработчики утверждают, что на 64-битной системе Cineerra работает более стабильно, чем на 32-битной.

Что до видеоплаты, то, не следуя примеру той же коммерческой продукции для Windows, Cineerra в этом плане особых требований не предъявляет — не потребуется даже поддержка OpenGL. Собственно, Cineerra требовательна только к трем вещам — скорости процессора, оперативной памяти и месту на жестком диске.

Интерфейс Cineerra основан на особом, разработанном в рамках проекта, движке, который больше подходит для нужд создателей Cineerra, чем GTK и Qt. Кстати, вы уже могли столкнуться с ним в продукте Broadcast2000 — предшественнике Cineerra.

Cinelegra — продукт нестабильный, так что запасайтесь терпением и почаще сохраняйтесь. Не стоит думать, что Cinelegra не пригодна для серьезной работы: есть люди, которые профессионально, на заказ, делают в Cinelegra клипы и рекламные ролики.

Однако сразу освоить Cinelegra не удастся. Это продукт, требующий чтения документации и, как показывает опыт, общения с другими пользователями. Навыки работы с Adobe Premiere вам мало пригодятся. В Cinelegra все совсем по-другому — иногда удобно, а иногда странно. Например, вы не можете просто так загрузить файлы, сжатые в MPEG-1 или MPEG-2. Вернее, можете, но тогда Cinelegra прочтет только небольшой фрагмент из такого файла. А чтобы подгрузить его целиком, надо сначала создать индексный файл. Делается это при помощи утилиты `mpeg3toc`, входящей в состав Cinelegra. Она предназначена для запуска из консоли и требует указания полного пути к имени MPEG-файла. Например, индексный файл `index13.toc` из фильма `movie13.mpg` получаем следующим образом:

```
mpeg3toc /mnt/videos/test/movie13.mpg index13.toc
```

Затем в Cinelegra загружаем не `movie13.mpg`, а именно соответствующий фильму индексный файл — в нашем случае это `index13.toc`.

Для DVD индексировать надо не VOB-файлы, а IFO: `mts_01_0.ifo` и подобные. А видео MPEG-4 индексировать не нужно, оно и так загружается нормально.

Cinelegra умеет также загружать (и экспортировать) последовательности изображений-кадров в форматах PNG, TIFF, TGA и JPEG. Однако если вы захотите загрузить в монтаж не один такой файл, а сразу много, то столкнетесь с еще одной особенностью Cinelegra — умением загружать по несколько файлов одновременно. Чтобы обойти эту преграду, надо использовать опять

На заметку домашнему видеоредактору

Плюсы и минусы Cinelegra

Сначала о плохом. Нестабильность этой программы стала уже привычным явлением. Но работать все-таки можно. Второй несколько раздражающий фактор — невозможность просто взять мышью кусочек видео и перенести его в другое место. Не умеет пока этого Cinelegra. Вы можете копировать фрагменты в произвольные места, смещать начало фрагмента относительно места его отображения, добавлять пустоту перед фрагментом, но взять и переместить — ни за что. Работа в Cinelegra больше похожа на склеивание видео из кусочков, в то время как подход Adobe Premiere — это, образно выражаясь, составление

мозаики, где каждый кусочек можно в любой момент без хлопот переставить. Теперь о хорошем. Требования Cinelegra к аппаратуре не столь жестки, как у ее аналогов для систем Windows и Mac OS. Функций у Cinelegra достаточно, чтобы вполне профессионально смонтировать клип или даже фильм. Если вы видели последний фильм Ингмара Бергмана «Сарабанда», то не могли не заметить, насколько просто и естественно выполнен там монтаж. То же самое можно сделать и в Cinelegra, хотя ее «выразительной мощности» хватит и для быстрых, насыщенных эффектами музыкальных видеоклипов.

же индексный файл, но уже особого (впрочем, довольно простого) формата. Утилиту для создания таких файлов можно скачать по адресу <http://malefico3d.com.ar/tutor/img2list-en.html>, или же самостоятельно написать простейший скрипт. Из видеформатов Cinelegra понимает также несжатый QuickTime.

Что касается звука, Cinelegra может работать с форматами WAV, PCM и AIFF, а также AC3 audio, OggVorbis и MP3. В последнем случае файлы с переменным битрейтом будут помехой (хотя они нормально воспроизводятся), поэтому лучше либо проиндексировать их, либо заблаговременно перекодировать в WAV.

Собственный формат «проектных» файлов Cinelegra — XML. Программа никуда не копирует используемые клипы и семплы, а в XML-файлах хранит ссылки на эти материалы.

Интерфейс у Cinelegra состоит из четырех основных отделенных друг от друга окон. «Resources» — своеобразная палитра, в которой доступны клипы, медийные файлы, эффекты и переходы. «Viewer» — просмотрщик выбранного видеоклипа, используется также для точной нарезки видео. «Program» — поле, в котором происходит монтаж звука и видео. Вам доступно практически неограниченное число дорожек. Cinelegra позволят применять к ним эффекты (как видео, так и звуковые). Каждая видеодорожка оснащена огибающими затемнения (fade), наплыва и так далее, то есть параметры этих эффектов можно регулировать динамически на протяжении отрезка времени. Аналог окна «Program» в Adobe Premiere называется «Timeline».

В поле «Program» можно выделить фрагмент и сохранить его в окно «Resource» как клип — разумеется, виртуально, без физического переноса медиаданных.

Наконец, еще одно окно — «Compositor». В нем отображается видео, посылаемое на выход из «Program» (с применением эффектов), и панель инструментов здесь почти такая же — чтобы удобнее было делать монтаж, имея перед глазами не длинную полосу, разбитую на кадры, а полноразмерное изображение.

Несколько слов напоследок

Мы рассказали о некотором наборе софта, который кажется нам оптимальным для работы с видео на платформе Linux. Очевидно, что Linux предоставляет достаточно средств для полноценной обработки видео, и, что очень важно, стоимость этих средств по сути равна нулю. Что до функциональности, в плане кодеков и кодировщиков софт под Linux дает фору прочему программному обеспечению (за исключением разве что многоплатформенного). По части нелинейного монтажа видео Linux все же отстает. Увы, крупные компании, производящие ПО для нелинейного монтажа, пока мало обращают внимание на Linux.

Однако ситуация начинает понемногу исправляться. К примеру, появилась Linux-версия редактора Main Actor (www.mainactor.com). Возможно, глядя на это, начнут подтягиваться и другие — сначала в области домашнего видеомонтажа, а потом уже портируют своих гигантов. Впрочем, тогда им придется вступить в острую борьбу за пользователей Cinelegra, Pitivi и других программ, и еще неизвестно, нужны ли будут к тому времени коммерческие аналоги. |

...ПЛЮС телефонизация всей Linux

Сотовые телефоны уже давно стали неотъемлемым атрибутом нашей жизни. Но фирмы-разработчики просто игнорируют существование альтернативных операционных систем. Они не выпускают программное обеспечение под Linux и не дают спецификаций.

Исключением является только фирма Nokia, которая выпустила свой SDK для разработки Java Midlets. Ситуация же складывается следующим образом: человек, купивший себе сотовый телефон, либо пользуется софтом под Windows, либо пишет свою программу для конкретной модели аппарата. В этой статье мы дадим несколько советов по использованию мобильных телефонов в связке с Linux и порекомендуем некоторые интересные программные продукты.

| Использование GPRS |

GPRS — это технология пакетной передачи данных по радиоканалу. Телефон здесь выступает в роли обычного модема, поддерживающего многие AT-команды. Следовательно, проблем с настройкой GPRS не должно быть ни у каких сотовых аппаратов. Работать можно как через дата-кабель, так и через инфракрасный порт или Bluetooth.

| Настройка ядра |

Для нормальной работы с GPRS мы должны установить пакеты протокола ppp (point to point):

```
#yum install ppp
```

Также необходимо собрать нужные модули ядра (в большинстве систем они собраны по умолчанию) для поддержки ppp:

Device drivers → **Networking Support** → **PPP (point to point protocol) support**

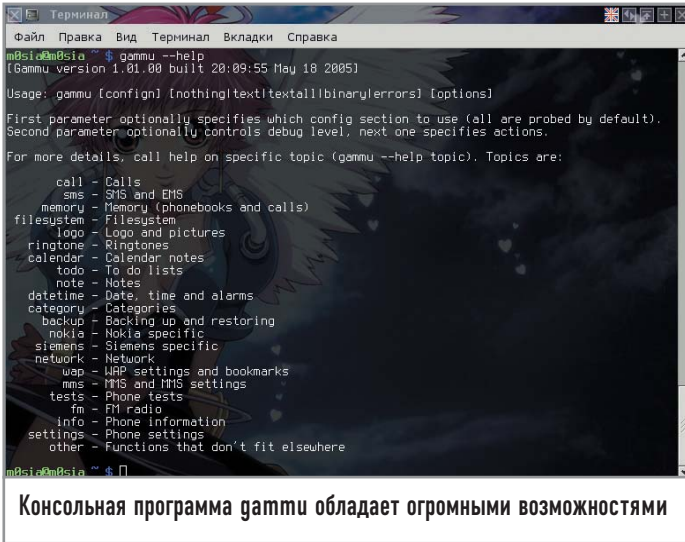
| Настройка Bluetooth |

Для использования GPRS через Bluetooth необходимо установить следующие пакеты:

```
#yum libbluez bluez-sdp bluez-pan bluez-hciemu bluez-utils
```

Для поддержки bluez (официального стека протоколов Bluetooth ветки ядра 2.6) мы должны включить несколько модулей ядра. В /etc/modules.conf нужно добавить строки, которые подключат модули поддержки Bluetooth:

```
alias net-pf-31 bluez
alias tty-ldisc-15 hci_uart
alias bt-proto-0 l2cap
alias bt-proto-2 sco
alias bt-proto-3 rfcomm
alias bt-proto-5 bnep
```



Теперь внесем в /etc/bluetooth/hcid.conf следующие строки:

```
options {
autoinit yes;
security user;
pairing multi;
pin_helper /usr/bin/bluepin; //программа выдает PIN-код
телефону по запросу
}
device {
name "workstation";
class 0x3e0100;
iscan disable; //устройство не видно при сканировании
pscan enable; //с устройством можно устанавливать связь
lm accept, master;
lp rswitch, hold, sniff, park;
auth enable;
encrypt enable; //соединение шифруется
}
```

Также нужна программа, которая по запросу выдавала бы PIN-код телефону. Для этого директива pin_helper в разделе «Options» должна выглядеть следующим образом:

```
pin_helper /etc/bluetooth/bluepin;
```

После этого мы можем протестировать работу Bluetooth. При правильных настройках получится следующий вывод:

```
#hcidtool scan
Scanning ...
00:12:62:0F:4E:2F Nokia 6230
```

Теперь проверим наличие DUN-сервиса для дозвона по GPRS через Bluetooth:

```
#sdptool search DUN
```

После этого создадим устройство:

```
#rfcomm bind 0 00:12:62:0F:4E:2F 1
```

Мы только что подключили устройство 0 к bluetooth-устройству 00:12:62:0F:4E:2F на первый канал. В том случае, если вы не используете devfs или udev, файлы /dev/rfcomm* могут отсутствовать. Поэтому нам потребуется их создать с помощью вот такой команды:

```
#mknod /dev/rfcomm0 c 216 0
```

На этом этап конфигурирования Bluetooth закончен.

Настройка IrDA

Для связи с телефоном использовался Tekram IrMate 210B dongle. Чтобы он корректно работал, необходимо настроить ядро, включив в него следующие модули:

```
Device Drivers ---> Networking support ->
<*> IrDA (infrared) subsystem support --->
<*> IrCOMM protocol
Infrared-port device drivers --->
<*> IrTTY (uses Linux serial driver)
[*] Serial dongle support
<*> Tekram IrMate 210B dongle
```

Также необходимо установить пакет irda-utils:

```
#yum irda-utils
```

После этого запускаем команду /usr/sbin/irattach /dev/ttyS0 -d tekram -s. Программа должна указать, что на COM-порту 1 находится инфракрасный датчик фирмы Tekram. В итоге получаем устройство под названием /dev/ircomm0.

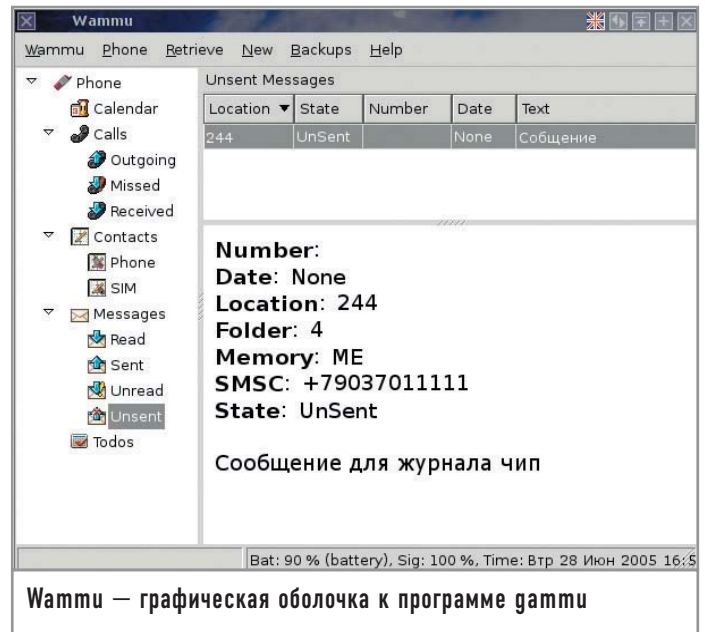
Настройка связи через USB/COM-кабель

Если вы счастливый обладатель кабеля, подключающегося к COM-порту, то никакой дополнительной настройки вам производить не надо, достаточно далее использовать устройство /dev/ttyS0(serial port 0).

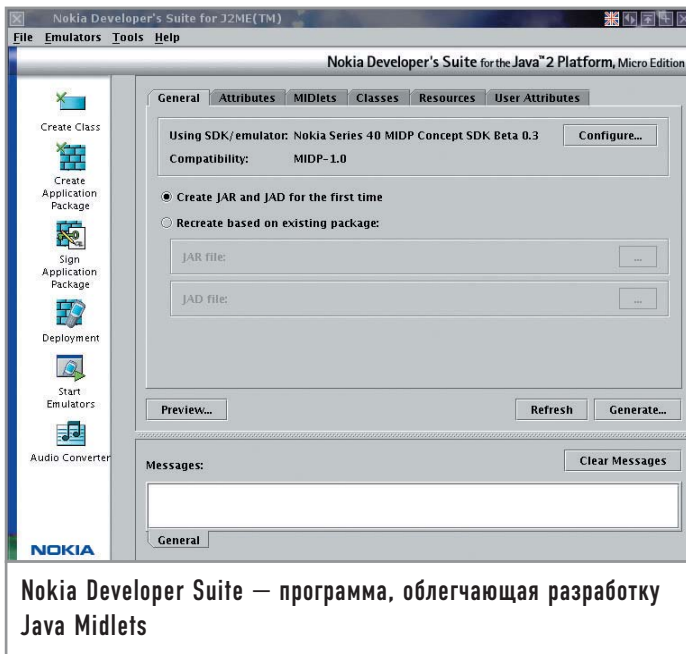
В том случае, если вы работаете с USB-кабелем, система hotplug, скорее всего, сама найдет нужный модуль и создаст устройство /dev/ttyUSB0 или /dev/tts/USB0. Если по каким-то причинам этого не произошло, вам придется попробовать самостоятельно собрать все необходимые модули из раздела конфигурации ядра «Device Drivers → Usb Support → USB Serial Converter support».

Настройка PPP

Настройка демона PPP одинакова для всех видов подключения. Мы продемонстрируем ее на примере подключения к сети «Билайн» через IrDA. Для этого создадим файл /etc/ppp/peers/beeline:



Wammu — графическая оболочка к программе gammu



Nokia Developer Suite — программа, облегчающая разработку Java Midlets

```
/dev/ircomm0 57600 noauth defaultroute usepeerdns updetach
persist
noipdefault lock connect 'chat -v -f /etc/ppp/beeline'
novjccomp norcomp noaccomp noipdefault nodeflate novj nobsd-
comp
```

В случае подключения через Bluetooth /dev/ircomm0 нужно заменить на /dev/rfcomm0, а при работе с кабелем использовать устройство /dev/ttyUSB0. После этого создадим chat-файл /etc/ppp/beeline:

```
TIMEOUT 45
ABORT "NO CARRIER"
ABORT ERROR
"" 'ATE1'
OK AT+CGDCONT=1,"IP","internet.beeline.ru"
OK ATD*99***1#
CONNECT c
```

В файл /etc/ppp/ppp-secrets добавляем следующую строку:
beeline ppp0 "beeline"

После этого достаточно набрать `pppd call beeline`, и мы в сети.

Работа с телефоном

Для работы с сотовыми телефонами под Linux написано великое множество различных программ. Единственный их огромный минус в том, что они создаются под конкретного производителя сотовых телефонов (а зачастую даже под конкретную модель). Ситуация усугубляется отсутствием единых стандартов. Единственная программа, которая способна работать с большим количеством самых разнообразных моделей телефонов, — это `gammu`. Список в той или иной степени поддерживаемых ею сотовых аппаратов достаточно велик. В первую очередь программа создавалась для работы с телефонами фирмы Nokia, поэтому может полностью заменить собой утилиту Nokia PC Suite.

Для настройки программы требуется создать файл `gammu.rc` в домашней директории. Для соединения по инфракрасному порту в него надо вписать следующие строки:

```
[gammu]
port = /dev/ircomm0
connection = irda
```

Для соединения через Bluetooth конфигурационный файл будет выглядеть так:

```
[gammu]
device = 00:12:62:0F:4E:2F
connection = bluephonet
```

Функций у программы огромное количество: получение и отправка SMS и MMS, бэкап содержимого телефона, работа с записной книжкой, файловой системой телефона (загрузка картинок, мелодий и игр), календарем, проведение тестов, редактирование FM-станций. Также с ее помощью можно выяснить код защиты телефона. Более подробно о большинстве функций вы можете узнать, набрав команду:

```
gammu -help
```

К `gammu` существует графический интерфейс — `Wammu`. Он позволяет читать SMS, работать с записной книжкой, редактировать календарь. К сожалению, больше никаких функций `gammu` в нем не реализовано, и в настоящее время проект не развивается.

Программирование для сотовых

Компания Nokia, понимая перспективность Linux как платформы, выпустила Nokia Delopment Kit. Эта среда может быть интегрирована с Borland Jbuilder, либо Sun JavaOne, либо существовать отдельным пакетом. Она включает в себя эмулятор сотовых телефонов Nokia серий 40 и 60 и редактор MIDI-файлов. Также в ней имеются примеры Midlets, которые помогут начинающему программисту создавать собственные утилиты для работы с сотовыми телефонами. |

Программное обеспечение

Полезные утилиты

Кроме упомянутых выше существует еще много специфических утилит для разных моделей телефонов. Расскажем кратко о некоторых из них.

Siemens

- ▶ **SieFs**. Виртуальная файловая система для доступа к памяти мобильного телефона Siemens. Позволяет работать с телефоном как с дискетой.
- ▶ **FlasherFlash для Linux**. Вытаскивает/заливает full-flash, eeprom. Поддерживает патчи V-Klay.

- ▶ **SCMxx**. Консольная программа для работы с телефонами Siemens. Позволяет загружать/удалять мелодии, логотип, записи адресной книги, а также отправлять и получать SMS.

- ▶ **gSCMxx**. Графическая оболочка для SCMxx.
- ▶ **obexFTP**. Аналог SieFs. Работает через IrDA, Bluetooth.
- ▶ **Smscmd**. Программа позволяет использовать телефон с поддержкой SMS как консоль *nix-систем. Работает в связке с SCMxx.

Motorola

- ▶ **Moto4Lin**. Файловый менеджер для Motorola p2k.
- ▶ **MIDlet loader**. Загрузчик Java Midlets на телефон.

Samsung

- ▶ **Samsung SGH-V200 photo utility**. Программа, способная «вытаскивать» фотографии из аппарата Samsung SGH-V200.

Александр Прокудин

Клеим панораму

Забираясь вверх по склону Ай-петринской яйлы, вы и подозревать не могли, что всего через каких-то пять метров окажетесь на вершине скалы, за которой начинается море. При виде восхитительного пейзажа пальцы сами собой начинают судорожно давить на кнопку спуска затвора оказавшейся как раз под рукой камеры. Но по возвращении домой выясняется, что сделанные снимки, какими бы резкими они ни были, не отражают целиком всю полноту картины, которую вам довелось увидеть. У вас словно украли часть пейзажа.

В принципе так и есть: соотношение сторон на фотоснимках не соответствует размеру поля, воспринимаемого зрением. По сути, фотографии оказываются прилично обрезанными справа и слева. Существует два способа решить эту проблему. Первый заключается в том, чтобы осуществлять съемку широкоугольным объективом, а затем в цифровой копии на компьютере отрезать лишнее сверху и снизу. Но этот вариант далеко не идеален, поэтому мы рассмотрим гораздо более перспективный метод — панораму.

Принцип ее создания чрезвычайно прост. Вы делаете один кадр, затем поворачиваете камеру вправо или влево — чтобы осталась часть пейзажа из предыдущего кадра, и делаете следующий снимок. Затем еще раз поворачиваете камеру — и так до тех пор, пока не добьетесь нужного результата.

Кроме того, панораму можно снимать в два-три ряда, а еще лучше — вертикальными кадрами. Правда, во втором случае штатив становится скорее помехой. Желательно, чтобы при съемке панорамы оптическая ось имела строго горизонтальное направление, а экспозиция была заблокирована.

Сделав серию таких снимков, их останется склеить на компьютере. Храбрые сердцем обладатели свободного времени могут делать это и вручную. Всего-то надо добавить один снимок к другому новым слоем, сделать оба слоя полупрозрачными, а затем, перемещая их относительно друг друга и вращая вокруг своей оси, подогнать до полного совпадения и отрезать лишнее. Опытные же люди для этого используют специальные программы.

Практичное решение

Принцип действия почти всех программ для создания панорам весьма прост. Программе нужно показать одинаковые фрагменты на двух снимках. Из них она формирует контрольные точки, по которым затем и склеивает две фотографии. Обычно рекомендуется указывать не меньше четырех пар контрольных точек для каждой пары изображений, с тем чтобы панорама не

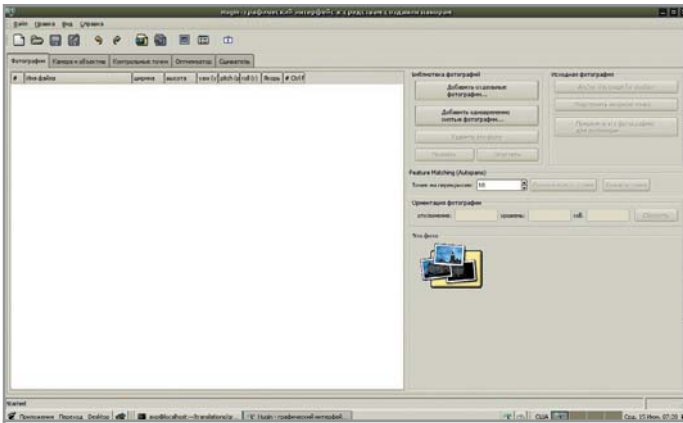
выглядела перекошенной. Этот принцип действует и в программе hugin, на примере которой мы и расскажем вам о том, как в GNU/Linux самостоятельно склеить панорамы. Вообще-то, hugin — это графический интерфейс к серии программ Panorama tools (подробнее об этой серии можно узнать по адресу: www.path.unimelb.edu.au/~dersch/). Помимо пакетов с исходным кодом (<http://hugin.sourceforge.net>) встречаются также бинарные пакеты для разных дистрибутивов.

Начало работы

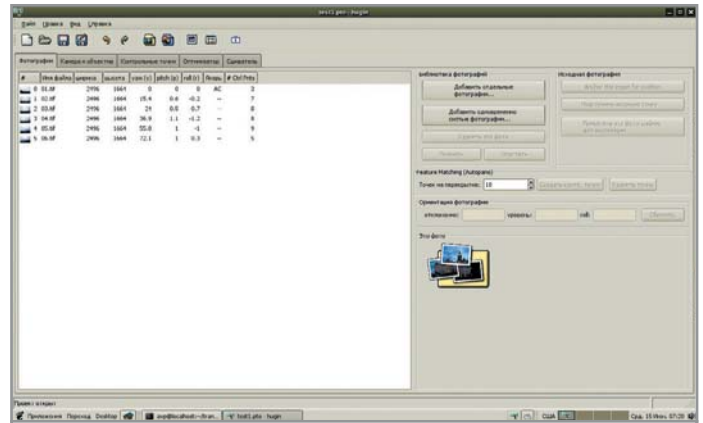
Запустите программу и добавьте в проект фотографии в таком порядке, чтобы первой была крайняя слева, а последней — крайняя справа. Затем перейдите с вкладки «Фотографии» на «Контрольные точки». В левой части откройте изображение с номером «0», в правой — с номером «1». Это будет первая пара фотографий.

На фотографии «0» укажите курсором в виде окружности максимально контрастный фрагмент снимка — например, какой-нибудь резкий выступ скалы на фоне моря. Подождите, пока программа пройдет по парной фотографии «1», чтобы найти на ней аналогичный фрагмент. Когда точка будет определена правильно, щелкните мышью внизу по кнопке «Добавить». Если вы всецело доверяете программе, просто выставьте справа флажок «Сразу добавлять точку». Добавив четыре пары контрольных точек, кликните по кнопке со стрелкой, указывающей направо, чтобы перейти к следующей паре фотографий, и повторите описанную выше процедуру еще раз.

Как только с добавлением минимального количества контрольных точек для всех пар будет покончено, переключитесь на вкладку «Оптимизатор». Здесь необходимо выбрать способ, при помощи которого программа совместит изображения по контрольным точкам так, чтобы исключить появление швов. Что касается способов, то они отличаются друг от друга лишь типом дополнительной коррекции. Например, если злой рок



Графический интерфейс к средствам создания панорам



Добавьте все необходимые изображения

настиг вас в момент съемки на коротком конце зума панорамы близко находящихся объектов, выберите способ оптимизации «Позиции с учетом бочки (у,р,г,b)». Программа сама тихо и мирно уберет «бочку», заботливо «сколоченную» вашими руками.

Пройдя этап оптимизации, переключитесь на последнюю вкладку «Сшиватель» и выберите, в файл какого формата и с какими параметрами вы хотите записать панораму. Можно указать тип панорамы (прямоугольная, цилиндрическая), рассчитать поле обзора, количество пикселей по горизонтали и вертикали, указать параметры постобработки — например, изменение экспозиции или гаммы.

Корректировка изображения

Теперь остается разобраться с еще одним непростым моментом. Мы живем в мире, где все постоянно движется: деревья от ветра, облака по небу, собачки, человечки... Пока вы меняли ракурс съемки, облачко переместилось и солнце выглянуло, следовательно, освещение поменялось, а значит где-то трава стала светлее, другое облачко — блее, ну и так далее. На панораме такая смена освещения будет выглядеть так, как будто кто-то осветлил ее по вертикали в несколько полос, причем границы между этими полосами будут видны невооруженным глазом.

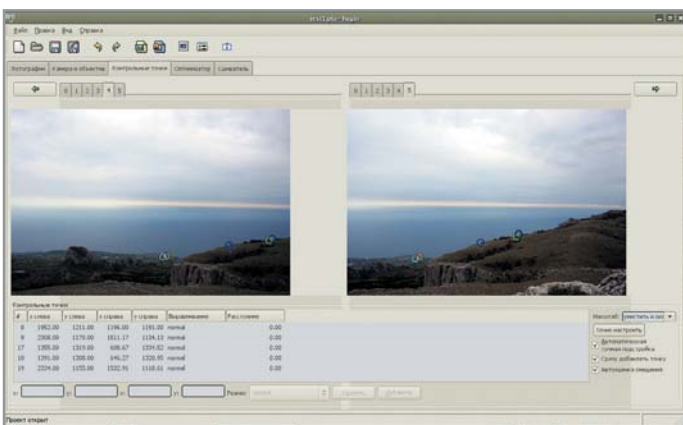
Для того чтобы сгладить этот эффект, hugin после склеивания проходит по результату программой enblend, делающей границы склеенных снимков плавно перетекающими друг в друга. Enblend можно скачать по ссылке <http://enblend.sourceforge.net>

forge.net и компилировать отдельно. При желании эту программу можно запускать из командной строки, но в нашем случае она попросту должна быть скомпилирована и установлена перед запуском процесса сшивки панорамы. Остается добавить, что при использовании Enblend на вкладке «Сшиватель» нужно указать значение растушевки равным нулю. Все готово? Тогда щелкайте по кнопке «Сшить!» и ждите результата.

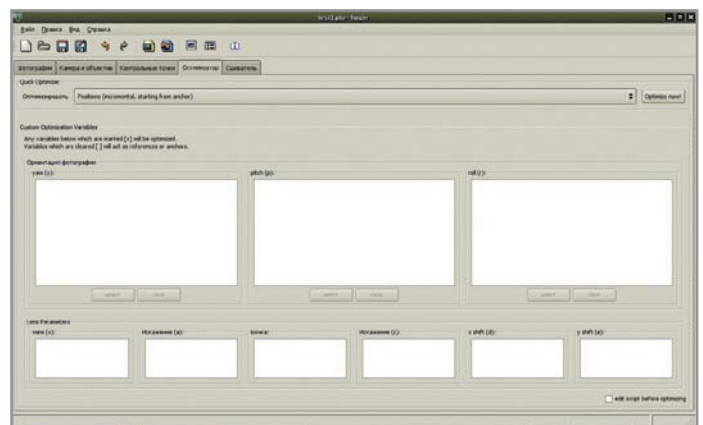
Кадрирование

По результатам сшивки панорамы вам однозначно захочется кадрировать полученное изображение, чтобы избавиться от неровных краев. В GNU/Linux вашим инструментом с большой долей вероятности окажется GIMP, который до серии версий 2.5-2.6 работать с 16 битами на канал не будет. Потеря части цветовой информации будет заметно ощутима в том случае, если вы на всех стадиях подготовки панорамы предполагаете работать с 16-битными TIFF (а значит, и снимать в RAW). Как только в производственной цепочке появляется JPEG, битность на канал автоматически, независимо от инструмента обработки, понижается до восьми — таковы ограничения формата JPEG.

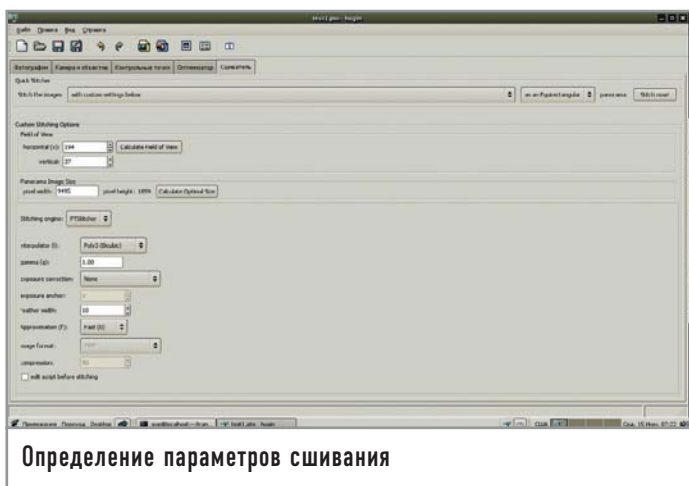
Поскольку разглядывать панорамы с экрана имеет смысл в случае, если у вас 23-дюймовый широкоформатный Apple Cinema, следующим шагом после сшивания будет запись файла на CD, который впоследствии отправится в фотолабораторию, сотрудники которой, в свою очередь, тоже при любом удобном случае помашут рукой вашему 16-битному снимку и сделают его 8-битным.



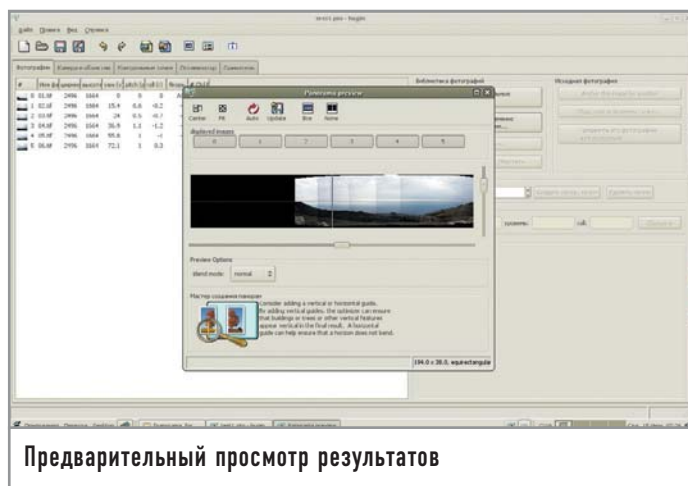
Обозначьте контрольные точки



Оптимизируйте изображение



Определение параметров сшивания



Предварительный просмотр результатов

Итого, если 16-битная обработка для вас принципиально важна (а ей все-таки не стоит пренебрегать), рекомендуем использовать свободный Cinepaint доступный по ссылке <http://cinepaint.sf.net>; коммерческий, с закрытым кодом Pixel32 (www.kanzelsberger.com, цена — от \$32 до \$100) или покорно ждать интеграции GIMP с GEGE, теперь уже точно назначенной на осень 2005 года. Пользователи GIMP 2.3-2.4 могут заметно упростить себе жизнь, попросив у сотрудников лаборатории ICC-профиль их принтера и подключив его через вкладку «Управление цветом» («Color management») в диалоге настройки приложения.

Другие варианты решения

И последнее. GNU/Linux не была бы такой популярной операционной системой, если бы на каждый vim не приходился свой Emacs. В случае с панорамами также всегда найдутся иные варианты решения задачи.

Во-первых, существует модуль для GIMP под названием Panorama stitch. Он работает при условии, что этот графический редактор собран с поддержкой расширений на Python. Проверить это просто: посмотрите в меню окна GIMP с любым изображением. Если увидите там Python-Fu — значит все в порядке. Если нет — просто пересоберите GIMP с ключом `--enable-python`. Имейте в виду, что для Panorama stitch нужен GIMP версий 2.2.x. Любые версии ниже 2.2, а также новые (от 2.3.0 и выше) для этого расширения пока (или

уже) не подходят. Скопируйте файл `stitch_0.9.5.py` в каталог `~/gimp-2.2/plug-ins/`. После перезапуска GIMP вы найдете это расширение в меню «Расширение → Утилиты» основного окна GIMP. К файлу сценария прилагается очень внятно написанная документация в формате PDF. Но поскольку принцип действия тот же самый, зарываться в нее с головой вам не придется. Только не забудьте пройтись по конечному файлу программой `enblend`.

Во-вторых, есть приложение `autopano-sift`, доступное по ссылке <http://user.cs.tu-berlin.de/~nowozin/autopano-sift/>. Оно написано на Mono/C# и предназначено для автоматического определения контрольных точек, а также прекрасно интегрируется с `hugin`. По сути, `autopano-sift` выглядит как один большой мастер создания панорам и рекомендуется новичкам и тем, кто не хочет вручную указывать `hugin` контрольные точки. Полученный в результате работы программы PTO-файл можно открыть как проектный файл `hugin`.

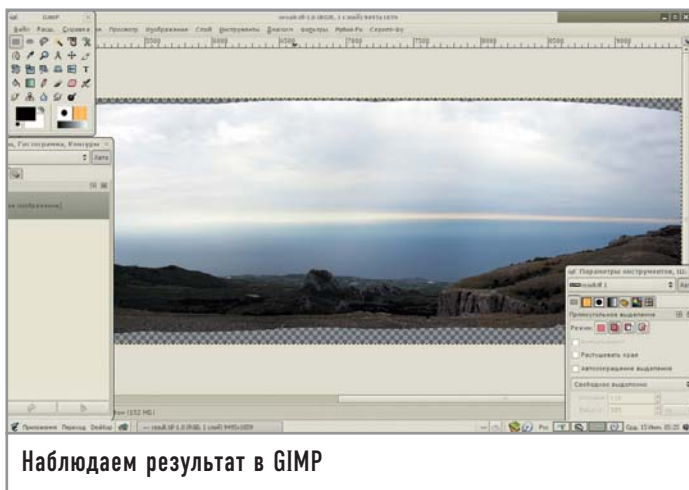
Напоследок хочется все-таки посоветовать еще раз как следует обдумать вопрос инвестиций в штатив. Нет ничего обиднее шевеленки в интересном кадре. |

Советы и примечания

К вопросу о производительности

Если панорама клеится из нескольких файлов в формате JPEG, когда размер каждого из них при всем желании никак не больше 3 Мбайт, то на время сшивки можно спокойно переключиться на соседний виртуальный рабочий стол и проверить почту или убить пару монстров (если успеете). На скорости работы почтового клиента или качестве прорисовки юнитов неприятеля это сильно не отразится. А вот если вы скормили `hugin` десяток полноразмерных 16-битных файлов в формате TIFF, луч-

ше просто уйти пить чай, чтобы не слышать надрывное жужжание жесткого диска. Тем, кто пользуется цифровыми зеркальными камерами со сменной оптикой, будет интересно поковыряться во вкладке «Камера и объективы», где можно указать массу параметров вроде угла обзора, фокусного расстояния при съемке, кроп-фактора и пр. `Hugin` умеет читать такие данные из EXIF-тегов ваших фотографий. Это в конечном итоге позволяет добиться более естественного вида панорамы.



Наблюдаем результат в GIMP

Плагиномания

Конец 2004 года принес нам два долгожданных события в мире Linux и свободного ПО в целом — выход финальных версий Mozilla Firefox (9 ноября) и Mozilla Thunderbird (5 декабря). По прошествии полугода уже можно говорить, что этот браузер стал одним из самых успешных проектов Open Source за всю историю.

Классическая Mozilla — интегрированный пакет для работы в Интернете, с честью исполнивший свою роль в истории. Сами разработчики не переставали говорить, что это не более чем новый движок, торопливо завернутый в газетную бумагу интерфейса Netscape 4, плюс поддержка вкладок. Словно рождественский пирог, Mozilla была нарезана на кусочки независимых приложений: браузер Firefox, почтовый клиент Thunderbird, редактор веб-страниц NVU, календарь-планировщик Sunbird и IRC-клиент Chatzilla (доступен как расширение). Популярность Firefox постоянно растет, версии 1.0.x загружены более 50 млн раз, программа становится серьезным игроком на рынке браузеров. В Финляндии доля Firefox перевалила за критическую отметку в 30%, после которой продукт перестал восприниматься как альтернатива. Другие страны также с завидным постоянством жонглируют двузначными числами, когда дело касается процентного соотношения присутствия этого браузера на рынке. Весьма любопытным фактом в связи с этим является то, что сама Microsoft, долгие годы утверждавшая, что Internet Explorer устраивает каждого, и не желающая дальше развивать свой браузер, опомнилась и взялась за разработку новой, седьмой по счету версии.

| Браузер-конструктор |

Не так давно Mozilla.org официально перестала поддерживать развитие интегрированного пакета Mozilla Suite, полностью сосредоточив усилия на развитии двух приложений — Firefox и Thunderbird. Остальные приложения развиваются сторонними компаниями. Например, HTML-редактор NVU поддерживается силами компании Linspire, выпускающей одноименный дистрибутив Linux-based. На эти события следует закономерная реакция создателей дистрибутивов Linux — в их детища теперь включаются Firefox, Thunderbird и NVU вместо старой доброй

Mozilla. Совместную жизнь интегрированного пакета и его молодых отпрысков резонно стараются не допускать.

Концепция продуктов Mozilla.org гармонично вписывается в общую структуру Linux: они гибко конфигурируются, масштабируются и похожи на конструкторы, где каждый пользователь может сотворить собственное неповторимое лицо и функциональный набор программы. Базовые дистрибутивы Firefox и Thunderbird не отягощены массой специфических функций, их реализация лежит на совести сторонних производителей — авторов расширений. Кроме того, правкой текстовых конфигурационных файлов можно также достичь заветной цели. Основные конкуренты Firefox и Thunderbird представляют собой монолитные продукты с традиционным подходом к конфигурированию через пункт меню «Настройка» или ему подобный.

Если в вашем дистрибутиве не оказалось Firefox и Thunderbird, их всегда можно загрузить из Интернета. Русские версии программ находятся на сайте Mozilla.ru Team. Переведенные на русский язык расширения находятся на форуме <http://forum.mozilla.ru> в разделах «Расширения» для каждой программы.

| Строительные кирпичики |

Описывать базовые возможности браузера и почтового клиента, рассказывать о миграционных проблемах, советовать, с чего начать, можно где угодно, только не в печатном издании, посвященном Linux. «RTFM», как говорят на форумах. Все это можно найти на русском языке в разделе «ЧаВо» сайта www.mozilla.ru.

Гораздо интереснее поговорить о многочисленных расширениях, раскрывающих истинную мощь продуктов Mozilla.org. Установка расширений в Firefox и Thunderbird немного отличается. В Firefox выберите меню «Файл → Открыть файл» и за-

тем в окне «Типы файлов» отметьте нижний пункт «Все файлы». Теперь откройте файл с расширением и перезапустите браузер. В Thunderbird эта процедура выполняется гораздо проще: «Инструменты → Расширения», затем нажмите внизу окна кнопку «Установить».

Расширения могут выполнять самые различные функции — управлять вкладками, добавлять кнопки/меню на панели, управлять загрузкой файлов, фильтровать рекламу, служить хорошим помощником в резервном копировании данных, а также осуществлять множество мелких сервисных функций. Расширений существует великое множество, и разобраться с нуля во всем этом красочном калейдоскопе становится весьма сложно.

Понимая, что новичкам будет довольно сложно освоить всю ветку форума «Расширения», Mozilla.ru Team регулярно выпускает Extension Pack — большой набор расширений для Firefox, объединенный общим удобным инсталлятором. Загрузить его можно также из раздела расширений <http://forum.mozilla.ru>. Файл называется `extrpack-<номер версии>.xpi`.

Сначала поговорим о наиболее интересных расширениях.

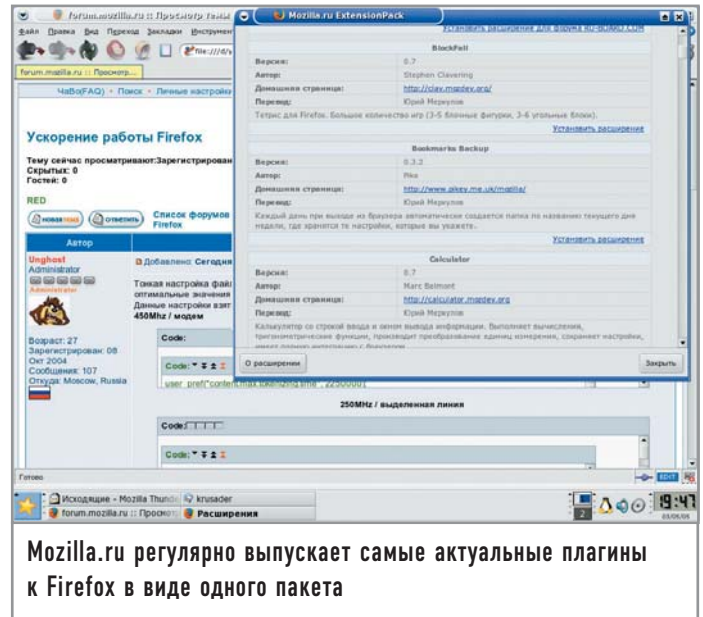
| Scrapbook |

Firefox имеет очень неприятную особенность — с его помощью невозможно открывать HTML-страницы с локального диска, если в пути/имени присутствуют символы кириллицы. Отчасти эта ошибка браузера сыграла немалую роль в росте популярности расширения Scrapbook, позволяющего каталогизировать ваши страницы на локальном диске, помещая их в веб-альбом, который отображается на боковой панели браузера. В контекстное меню браузера добавляются новые пункты, связанные с сохранением страниц, — «Сохранить» и далее по контексту. Эти пункты легко отличить от других, так как им присваиваются разноцветные значки. Например, для сохранения HTML-страницы — оранжевый, для сохранения файла — зеленый и т. д. Сохранять можно не только HTML-страницы целиком, но и кусочки страниц — текст, файлы, графику, выделенные фрагменты. Расширение имеет встроенный HTML-редактор, позволяющий из уже сохраненных страниц вырезать лишние куски — баннеры, меню, графику и т. д.

+ *Эlegantное, удобное средство сохранения и управления страницами на локальном диске, имеющее множество сервисных функций.*
 — *В случае использования другого браузера извлечение данных из веб-альбома представляет собой долгую и нудную работу, так как физически заголовки страниц на диске являются лишь цифровыми идентификаторами.*

| Adblock |

При веб-серфинге многочисленная реклама чаще всего представлена баннерами, кнопками, роликами. Она раздражает, расходует трафик, и ее желательно как-нибудь скрыть, не загружать на ваш компьютер, экономя нервы и драгоценное время. Можно указать Firefox загружать изображения только с оригинального сайта, но это не решает проблему в полной мере. Для стопроцентной блокировки рекламы создано расширение Adblock. В качестве настроек расширения заносятся некоторые подстроки. Если в адресе (ссылке) встретится какая-либо подстрока из настроек, загрузка любой информации с этого сайта будет запрещена.



Mozilla.ru регулярно выпускает самые актуальные плагины к Firefox в виде одного пакета

Настроить расширение можно вручную или воспользоваться готовыми фильтрами. Загрузите какую-нибудь страницу, найдите на ней элементы рекламного характера. В контекстном меню есть пункт «AdBlock изображение». Если вы его выберете, то изображение пропадет со страницы и в будущем никогда не будет загружаться. Всякий раз вручную убирать рекламу на каждой странице — дело хлопотное и не совсем благодарное. Гораздо эффективнее прописывать не полностью адреса, а только ключевые слова, которые встречаются в рекламе. Например, «banнер». Пропишите его качестве фильтра, и любая картинка с этим словом в адресе загружаться не будет.

Но можно пойти еще дальше, так как расширение поддерживает импорт фильтров из текстовых файлов. Выберите «Инструменты → Реклама (AdBlock) → Свойства». В открывшемся окне выберите «Настройки AdBlock → Импорт фильтров». Теперь просто укажите текстовый файл, в котором хранится массив подстрок, и фильтры окажутся в настройках расширения.

- + *Возможность стопроцентной фильтрации рекламы.*
- *Отсутствие каких-либо фильтров по умолчанию.*

| Tabbrowser Extensions |

TBE — одно из самых известных расширений для Firefox и Mozilla. У многих оно входит в «джентльменский набор» и сразу после установки Firefox (Mozilla) автоматически ставится TBE. Одно время даже выходила специальная модификация Firefox под названием BlackDiamond, фактически представляющая собой Firefox + TBE + небольшой тюнинг в виде измененных настроек по умолчанию. О TBE постоянно ведутся споры, но одно отрицать нельзя — это самое известное расширение в мире Mozilla.

Существует мнение, что TBE замедляет работу браузера. Это не совсем так. TBE замедляет скорость загрузки браузера, так как в память загружается довольно много дополнительного кода. На работу движка Gecko ни одно расширение не может влиять принципиально.

Расширение устанавливается так же, как и все остальные, каких-то своих особенностей у него нет. После перезагрузки

браузера будет предложен выбор: «Обычные настройки» или «Облегченные настройки». Особого значения это не имеет, так как эти настройки отличаются несколькими второстепенными пунктами, к тому же их всегда можно будет поменять позже.

Посмотрите на панель меню. У вас появился пункт «Вкладки». В нем собраны все функции ТВЕ (не настройки, а именно функции), которые дублируются правой кнопкой мыши на панели вкладок. Так что, когда все устанет — то есть ТВЕ будет идеально настроено под вас, — этот пункт можно будет скрыть встроенными средствами. Кроме того, в списке возможных кнопок для панели инструментов появятся три новые: «Список вкладок», «Загрузить набор вкладок» и «Недавно закрытые вкладки».

Настройки вызываются из меню «Вкладки → Настройки Tabbrowser Extensions». Если меню «Вкладки» скрыто, настройки можно вызвать стандартными средствами — через менеджер расширений.

Настройки представлены стандартными раскрывающимися списками (как, например, в «Настройке своего рабочего стола» KDE). Разобраться в огромном количестве пунктов меню, как ни странно, довольно просто, так как все настройки интуитивно понятны и тесно связаны между собой.

Все остальные расширения для работы с вкладками в Firefox лишь частично повторяют функциональный набор ТВЕ. Поэтому подробное описание данного расширения позволяет в полной мере представить все разнообразие настроек вкладок.

+ «Все в одном». ТВЕ заменяет множество других расширений, а также обладает несколькими уникальными функциями, которые до сих пор не реализованы ни в одном другом расширении, а кроме того, имеет удобный выверенный интерфейс, работает как в Firefox, так и в Mozilla.

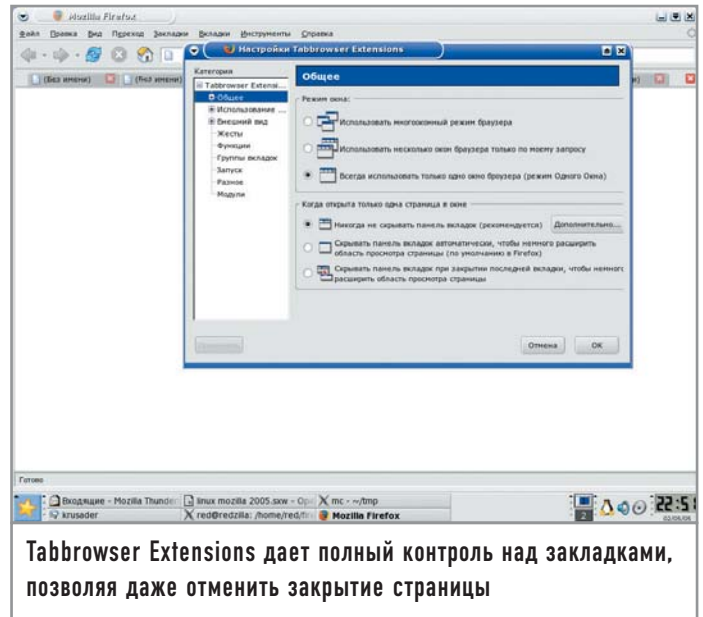
— Очень тяжелое расширение. Много повторяющихся настроек. ТВЕ отключает некоторые стандартные настройки и конфликтует с расширениями, имеющими аналогичные функции. Файлы конфигурации засоряются множеством настроек, причем после удаления ТВЕ дополнительные строки остаются. Очень чувствительно к версии браузера: чуть более новая или старая версия (сборка) — и ТВЕ не работает или работает с многочисленными ошибками. Поэтому перед его установкой обязательно проверяйте, подходит ли версия ТВЕ вашей версии браузера.

Дополнительные расширения

Попурри из расширений

Помимо уникальных расширений для браузера и почтового клиента существует группа универсальных расширений, работающая в обоих типах программ. Например, Mouse Gestures включает поддержку «мышиных» жестов в Firefox и Thunderbird (впервые были представлены широкой публике в Opera), QuickNote добавляет удобную запис-

ную книжку со вкладками. Расширения на сегодняшний день исчисляются сотнями, и уже сейчас можно смело сделать вывод, что их возможности у Firefox и Thunderbird практически безграничны. Одних только игр для Firefox написано уже с десяток. Например, Cards — большой набор пасьянсов, мечта любой секретарши.



Tabbrowser Extensions дает полный контроль над закладками, позволяя даже отменить закрытие страницы

| Flashgot |

В Linux на сегодняшний день поддерживается лишь одна программа — это Downloader for X, которую можно загрузить по адресу www.krasu.ru/soft/chuchelo, тем более что она входит во многие дистрибутивы Linux.

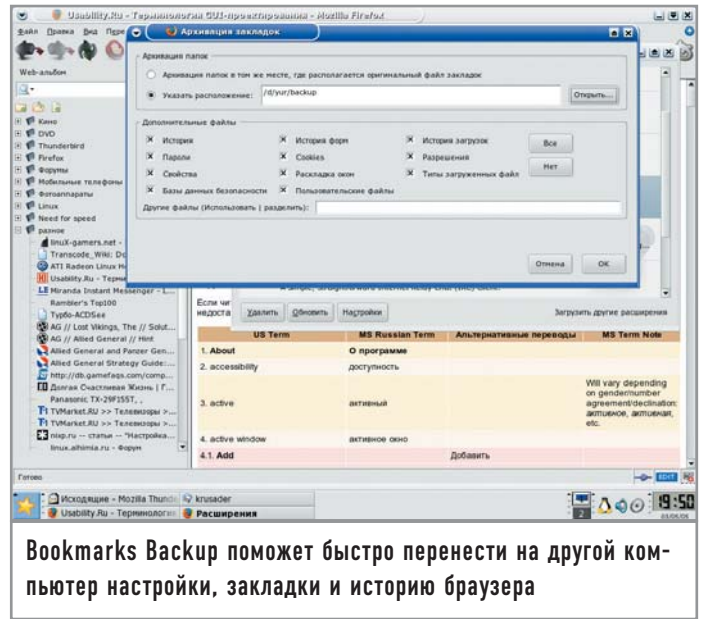
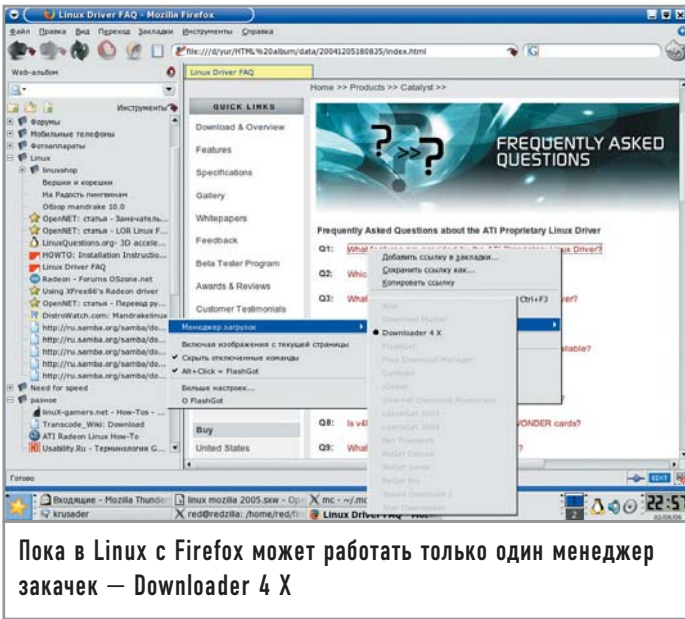
После установки расширения в контекстном меню Firefox появляется несколько новых пунктов: «Загрузить с помощью Flashgot», «Загрузить все с помощью Flashgot», «Создать галерею». Первый пункт наиболее часто используется, он просто передает адрес в качестве аргумента менеджеру загрузок, который принимается за работу. Второй пункт меню передает все ссылки со страницы в качестве аргумента. А третий позволяет создавать галереи из изображений текущей страницы.

+ Удобный инструмент для интеграции внешнего менеджера загрузок.

— Поддержка только Downloader for X в Linux, то есть отсутствие альтернативы при выборе менеджера загрузок.

| Bookmarks Backup |

Среди программ, создающих резервные копии пользовательских данных, стоит выделить данное расширение как наиболее простое и стабильное. Несмотря на скромное название, оно умеет создавать резервные копии практически всех пользовательских данных. Исключение составляют лишь расширения, которые в случае краха профиля браузера все равно придется переставить (но настраивать их не требуется — настройки расширений также сохраняются). По умолчанию расширение создает резервные копии внутри профиля (о том, что это такое и где он расположен, будет сказано позже), но разумнее держать копии в более безопасном, произвольном месте. В каталоге резервных копий создаются подкаталоги с именами дней недели, в которых уже находятся пользовательские данные. Из этого можно сделать вывод, что расширение хранит копии за последние семь дней. В случае краха профиля браузера (причин для этого может быть множество) нужно просто переписать содержимое последнего по дате резервного каталога в каталог профиля.



- + Простой удобный инструмент для резервирования пользовательских данных.
- Невозможность резервного копирования установленных расширений.

Теперь перейдем к расширениям почтового клиента Thunderbird.

| Outbox |

Простое расширение, которому давно уже пора поселиться в базовой сборке Thunderbird. Дело в том, что у почтового клиента нет на панели инструментов кнопку, выполняющую функцию отправки сообщений из папки «Исходящие», а во время составления сообщения вы не найдете кнопку «Отправить позже». Это очень неудобно, приходится каждый раз обращаться к меню, что занимает немало времени даже при средних объемах корреспонденции. Outbox добавляет эти кнопки на панели. Кроме того, вместо двух кнопок «Получить сообщения» и «Отправить сообщения» можно разместить одну «Отправить/получить сообщения», что, согласитесь, намного удобнее.

- + Добавляет жизненно необходимые функции.
- Претензия к разработчикам Thunderbird: давно пора включить это расширение в базовый дистрибутив.

| Tagzilla |

Расширение совершенно иного плана, позволяющее вставлять в письма случайные подписи. Для своей работы оно требует библиотеку jslib, которая ставится так же, как и расширение (<http://jslib.mozdev.org>). В настройках вы указываете неизменяемую часть подписи, а также файл, из которого будут браться строки для случайного выбора. Чаще всего в качестве таких строк выступают известные афоризмы или просто смешные фразы. После этого можно указать, автоматически или вручную расширение будет ставить подпись и открывать ли при этом окно расширения, чтобы иметь возможность ручного выбора строки из базы. Из дополнительных возможностей стоит отметить функцию извлечения строк из писем и помещения их в файл вашей базы.

- + Элегантный способ выделиться, «бесплатно» оживить свои письма.
- Расширение в большей степени создано для баловства и имеет проблемы с работой в Thunderbird 1.0.

| Quote Colors |

Настройка и раскраска цитирования всеми цветами видимого диапазона. Можно менять оттенок фона и цвет шрифта в зависимости от уровня цитирования, а также толщину и стиль линий цитирования, их местоположение (слева, справа, сверху, снизу — причем линии можно комбинировать: если, например, выбраны все четыре пункта, то получается рамка).

- + Очень красиво раскрашивает и оформляет письма, наглядно отображает цитаты.
- Давно не обновляется.

| Quick Text |

С этим расширением в Thunderbird можно создавать до десяти макросов — клавиатурных сокращений, которые будут превращаться в слова или фразы. В макросах можно использовать переменные.

- + Помогает при постоянном наборе сообщений, где встречаются одни и те же термины, особенно если для их набора требуется перекладывать раскладку клавиатуры.
- Всего лишь десять макросов, только «Alt»+<цифра>.

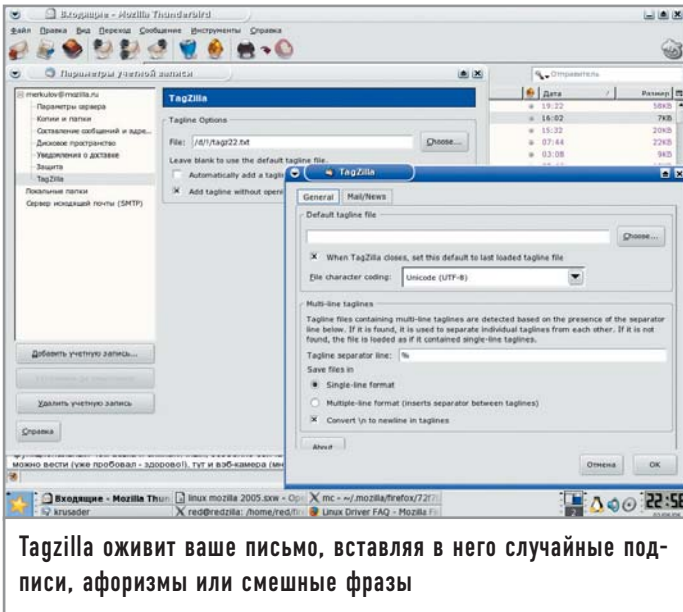
| Конфигурирование вручную |

Настройка продуктов Mozilla.org под себя не ограничивается одними лишь расширениями. Существует также возможность правки текстовых конфигурационных файлов, которые располагаются внутри профиля. Таких файлов три:

- ▶ user.js — в корневом каталоге профиля;
- ▶ userChrome.css — в подкаталоге /chrome;
- ▶ userContent.css — в подкаталоге /chrome.

Профили программ расположены в домашнем каталоге:

- ▶ Firefox — /.mozilla/firefox/<имя профиля>;
- ▶ Thunderbird — /.thunderbird/<имя профиля>.



Tagzilla оживит ваше письмо, вставляя в него случайные подписи, афоризмы или смешные фразы

В `user.js` устанавливаются разные свойства браузера. Часть из них описана на сайте <http://mozdev.org>. Все значения переменных могут быть изменены и через `about:config` в строке ввода адреса. Есть только маленькая тонкость. Когда вы создадите новый профиль или переустановите Firefox или Thunderbird, вам будет необходимо заново вводить все эти значения. При использовании `user.js` достаточно будет всего лишь перенести этот файл.

В `userChrome.css` вы можете редактировать отображение основных элементов браузера, стили, создать свой собственный стиль Firefox и Thunderbird. Обратите внимание, что файл `userChrome.css` не существует по умолчанию. После установки Firefox/Thunderbird вы найдете `userChrome-example.css`, содержащий примеры некоторых настроек, который можно переименовать в `userChrome.css`. Он предназначен для переопределения стилей страниц, которые загружаются в ваш браузер. Файл `userContent.css` также не существует по умолчанию (вместо него `userContent-example.css`).

Множество примеров настроек текстовых конфигурационных файлов можно найти в разделе «ЧаВо» на сайте www.mozilla.ru.

Перспективы развития

Все это множество расширений и настроек демонстрирует гибкость и мощь продуктов, созданных Mozilla.org. На дворе 2005 год, уже с успехом были выпущены финальные версии продуктов, и встает закономерный вопрос: а что же будет дальше? Основные планы разработчиков простираются к версии 2.0.

К выходу версии 1.1 летом у Firefox и Thunderbird будет поновому организовано меню настроек, в нем появятся вкладки. Кроме того, будут усовершенствованы система обновления программ и менеджер расширений. В Thunderbird появится проверка орфографии на лету — с подчеркиванием красной волнистой чертой слов с ошибками, как во многих текстовых процессорах. Станет возможным обновление `myspell` до версии, присутствующей в текущем варианте OpenOffice.

К версии 1.5 осенью программы обзаведутся многоязычными инсталляторами, новым встроенным менеджером загрузок (Firefox). Появится возможность перетаскивать мышкой панели инструментов — как, например, во всех программах рабочей среды KDE.

К версии 2.0 в конце года планируется внести встроенную поддержку переназначения горячих клавиш, новую систему разграничения прав, новый менеджер паролей. В Firefox появятся встроенные сервисы для глобального поиска, изменится работа с закладками и вкладками. Будет налажено взаимодействие с проектом Lightning для интеграции календаря с Thunderbird.

Ни Mozilla.org, ни конкуренты не спят: параллельно развиваются программы из состава KDE — браузер Konqueror и почтовый клиент Kmail, платный интегрированный пакет Opera, программы из лагеря GNOME — Eiphanh и Evolution. Каждая из них имеет свои особенности и круг сторонников, что в мире Linux давно уже стало обычным явлением. Но лишь программы Mozilla.org принесли в Linux воистину массовый, популярный набор инструментов для работы в Интернете. Не стоит также забывать и о том, что Mozilla.org создает кросс-платформенные приложения. Windows, Linux, Mac OS — везде вы увидите одни и те же Firefox и Thunderbird, а это значит, что, освоив их в одной ОС, вы без труда сможете выполнять те же самые задачи и в другой. Учитывая, что основное пополнение сообщества пользователей Linux происходит за счет миграции с других систем, отсутствие необходимости в изучении нового ПО для работы в Интернете становится неплохим козырем. В этом свете Firefox и Thunderbird являются универсальными и надежными проводниками в бескрайний мир Глобальной сети. |

Безопасность

Есть ли дыры в Firefox?

Как и предсказывали недоброжелатели, как только весть о Firefox разнеслась по всему миру и он поселился на миллионах жестких дисков пользователей Сети, так сразу в нем стали находиться одна за другой уязвимости. Тут следует сделать одну оговорку. Само по себе обнаружение уязвимости — это не более чем антиреклама для программного продукта, бьющая в случае с Firefox по его имиджу «самого безопасного браузера». Вторым этапом в подобных ситуациях принято считать написание и использование программ, которые эти уязвимости используют в деструктивных целях. Масовые эпидемии, связанные

с заражением компьютеров с установленной на них операционной системой Windows через почтовый клиент Outlook Express и браузер Internet Explorer — наиболее яркий тому пример. Но, к счастью, пока неизвестны реальные случаи заражения компьютеров через уязвимости в Firefox и Thunderbird. Так вот, возвращаясь к основной теме, стоит отметить тот факт, что эти дыры пока носят лишь теоретический характер, и дальше «лабораторий» проблема не распространяется. Программисты Mozilla.org оперативно латают найденные бреши, выпуская новые версии в рамках стабильной ветви разработки.

Юрий Меркулов

Контора пишет... считает и рисует

Linux долгое время росла сиротой — в ней не было достойного офисного пакета. Равно как поезд не может ехать без рельс, так и жизнь на десктопе немислима без хорошего офисного пакета. Свято место пусто не бывает, и на эту роль был приглашен пакет OpenOffice.org.

| Первый взгляд |

Сегодня стабильная ветка развития OpenOffice.org располагает уже четвертым релизом 1.1.4. Но на подходе совершенно новый продукт — OpenOffice.org 2.0. Пока выпущена только бета-версия, но по ней уже можно оценить функциональность предлагаемого продукта.

Версию OpenOffice.org 2.0 можно назвать колоссальным скачком вперед по сравнению с веткой 1.1. Это становится заметно с первых секунд, когда вы запускаете установку офисного пакета. Сразу же открывается просторный полигон выбора компонентов для установки. Мгновенно бросаются в глаза позитивные изменения: офисный пакет обзавелся программой для работы с базами данных, которая раньше была только в Star Office, средствами синхронизации с карманными ПК и другими мелкими вспомогательными утилитами. После установки можно заметить, что у всех программ сменились значки, так же как и у документов для каждого из компонентов офиса.

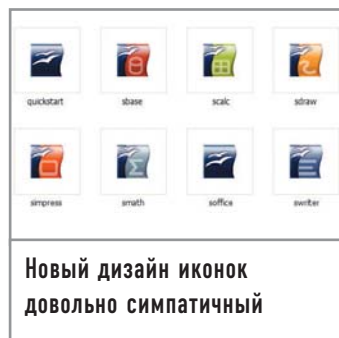
| Скорость |

OpenOffice.org 2.0 загружается немного быстрее предыдущего релиза, демонстрируя симпатичную заставку. На сайте <http://oosurvey.gratismania.ro/polls/> проходило открытое голосование по поводу выбора лучшей заставки для новой версии

пакета, так что даже в таком, казалось бы, пустяковом вопросе работа ведется фундаментально и сообща. Если используется QuickStart, то приложения буквально выстреливают на рабочий стол, задержка практически не ощутима на глаз. OpenOffice.org 1.1.x не обеспечивал такой скорости загрузки, а на слабых машинах польза от QuickStart вообще сводилась к нулю. OpenOffice.org 2.0 проворно открывает небольшие документы в «родном» формате, но по-прежнему невыносимо долго обрабатывает большие и сложные документы Microsoft Office. Значительно быстрее стали открываться новые окна и закрываться документы, невооруженным глазом заметно увеличение скорости работы интерфейса.

| Интерфейс |

Внешний вид OpenOffice.org преобразился до неузнаваемости — ему мог бы позавидовать любой здравомыслящий хамелеон. Подобно этому уникальному живому существу, OpenOffice.org 2.0 умело маскируется в той среде, в которую его установили. В Windows XP с включенной стилизацией окон OpenOffice.org 2.0 не отличить от Microsoft Office 2003 благодаря аналогичным графическим «выпуклостям» панелей инструментов. Классическая схема Windows XP придает OpenOffice.org 2.0 такой же классический вид. В KDE наш хамелеон прикидывается приложением QT, а в GNOME блистательно играет роль

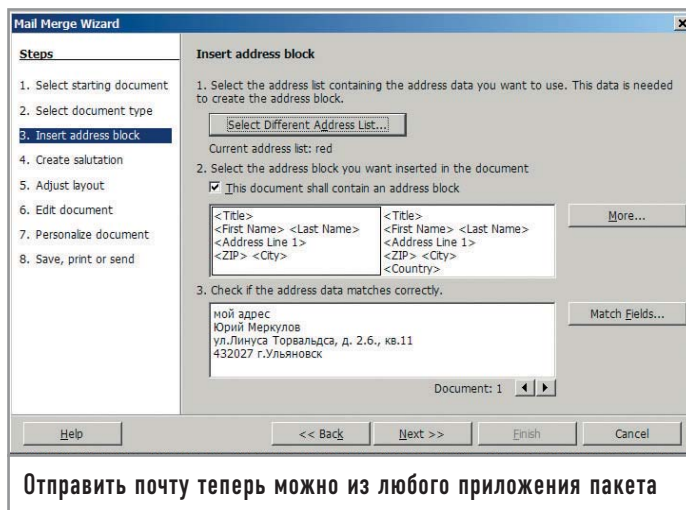


Gtk2-приложения. Но все же OpenOffice.org 2.0 по-прежнему не использует ни Gtk2, ни QT.

Замолвим словечко о панелях инструментов, которые теперь можно «таскать» за левый край по всему рабочему экрану приложения, как это принято делать в KDE или Microsoft Office. Если раньше, к примеру, при работе с таблицей нижняя строка панели «Форматирование» заменялась «Таблицей», то сейчас новая панель по умолчанию появляется в виде дополнительного окошка, которое можно прикрепить в произвольном месте. Закончили работу с таблицей, нет необходимости лицезреть ее панель — она автоматически скрывается. По умолчанию теперь нет боковой панели инструментов. Часть ее функций переключалась в верхние панели, а часть стала пунктами меню программы. Настройка панелей теперь не вызывается правой кнопкой мыши, а находится в меню «Tools → Customize → Menus». На первый взгляд, это откат назад. Но процедура настройки стала намного проще: все иконки представлены в списках, достаточно только выбрать необходимые. В OpenOffice.org 1.1.x далеко не все функции были по умолчанию укомплектованы значками, приходилось вручную создавать новые элементы панели инструментов.

Совместимость

Несомненно стоит отметить, что в OpenOffice.org 2.0 теперь появилась поддержка OASIS OpenDocument XML format, который недавно был утвержден в качестве стандарта Европейской комиссией. Конечно, никто не собирается законодательно обязывать использовать OpenDocument как формат официального документооборота, в отличие, например, от чертежей: для себя можно рисовать как угодно, а в работе все-таки лучше придерживаться стандартов. Но здоровая тенденция налицо: мир старается уйти от закрытого, кишашего вирусами (слово «небезопасный» не отражает и сотой доли проблем) формата, используемого в Microsoft Office. Первые ласточки вспорхнули еще в прошлом году, когда одна из крупнейших отечественных компаний, Сбербанк России, полностью запретила использование формата DOC в официальном документообороте, внедрив открытый, безопасный, но уже устаревший RTF. Формат OpenOffice.org 1.0 не является открытым, несмотря на свободный статус программы. Разработчики так и не создали отдельную документированную библиотеку, используя которую, можно было бы в любой программе открывать документы формата. В связи с тем что OpenOffice.org 2.0 переориентируется на OpenDocument, ждатель энтузиазма в продвижении старого формата не стоит.



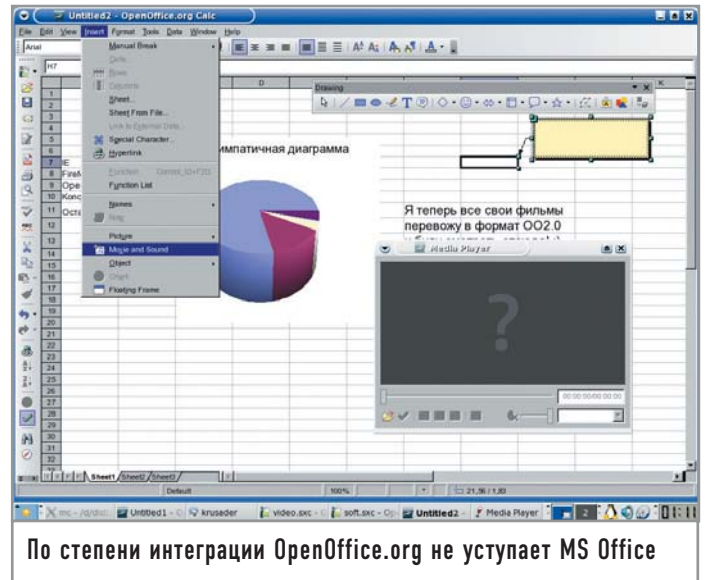
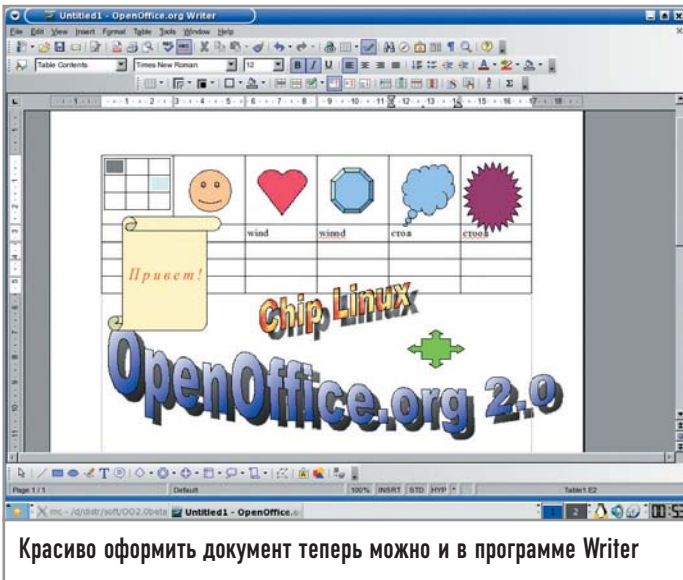
Поддержку же OpenDocument в своих продуктах может осуществить каждый, так как формат имеет открытые спецификации. Физически он обладает той же XML-структурой, что и OpenOffice.org 1.0 (sxw, sxc и т. д.). Мы попробовали сохранить с десяток текстовых документов и таблиц в новом формате и заметили, что файлы документов получаются на 5–10% объемнее, чем в OpenOffice.org 1.0. Это не страшно, если учесть, что документы Microsoft Office в 5–10 раз больше расходуют дисковую память.

Документы Word и Excel открываются без проблем, но, как уже говорилось выше, очень медленно. 10-мегабайтный файл DOC с таблицами и картинками загружался около 20 секунд на Barton 2500+ с объемом оперативной памяти 1 Гбайт. Честно говоря, мы уже начали думать, что офис завис, так как после пяти секунд работы индикация многозначительно замерла. Но если раньше даже такие нервные ожидания не гарантировали отсутствия катастрофической потери форматирования в документах, то теперь наконец-то все MSO-документы открылись в приемлемом виде. Где-то чуть-чуть съезжали картинки, где-то таблица выехала на сантиметр за поля, но ничего непоправимого не произошло. Совместимость с MSO улучшилась. Очень хочется верить в то, что и у вас также не возникнет серьезных проблем с открытием документов MSO, ведь даже разные версии этого офиса порой некорректно открывают собственные документы.

Конвергенция приложений

В любые документы OpenOffice.org 2.0 теперь можно вставлять не только графику, но и звуки, видео. В офисный пакет встроен собственный медиаплеер для проигрывания контента не выходя из программы. Реальную ощутимую пользу от этого нововведения получит OpenOffice Impress, так как презентации теперь могут превращаться во что угодно — от просмотра годового отчета компании по выпуску смычков под Лунную сонату Бетховена до презентации новой модели автомобиля под фильм «Такси».

Одним кликом теперь можно послать документ по почте — на панели красуется кнопка с изображением летящего конверта. Из меню «File» предлагается широкий выбор действий: послать просто как документ, как PDF-вложение, как HTML-



страницу. В меню «Tools» появился заботливый Wizard, который поможет новичкам отослать письмо с текущим документом. Не спешите падать со стула от смеха и умиления: письмо может быть как электронным, так и бумажным. Во втором случае Wizard поможет быстро и грамотно создать как само письмо, так и конверт к нему, в котором вы собираетесь послать подготовленный документ. Адрес берется из адресной книги, которая также является частью OpenOffice.org 2.0. Кроме того, новая версия имеет встроенную систему защиты документов цифровой подписью.

Улучшился экспорт в широко известный формат PDF. В свете того что большинство документации в качестве конечного формата использует именно этот формат, можно предположить, что OpenOffice.org 2.0 удастся немного потеснить легендарный TeX (LaTeX). Поддержка PDF появилась в уже с OpenOffice.org 1.1, однако в новой версии предлагаются более широкие настройки экспорта этого формата — вплоть до указания процента сжатия. Кроме того, есть возможность создавать PDF-документы с гиперссылками, которые могут указывать как на локальные страницы в документе, так и на ресурсы в Интернете. Экспорт в PDF в OpenOffice.org 2.0 предоставляет новые настройки сжатия JPEG-файлов. В предыдущих версиях пользователь мог выбирать только между тремя степенями DCT-сжатия, теперь же появился выбор между DCT- или LZW-сжатием без потерь.

В OpenOffice.org 2.0 имеется обновленный каталог графических фигур — Custom Shapes, во многом аналогичный каталогу Auto Shapes из пакета Microsoft Office. Огромный выбор фигур производит сильное впечатление, а рисование сложных схем перестало быть неразрешимой проблемой. Внутри каталога графических фигур несложно найти FontWork Gallery — аналог WordArt. По своим возможностям утилиты приблизительно одинаковы, но FontWork Gallery можно использовать в любом типе документа OpenOffice.org.

Отныне все делается значительно проще: можно мышью выделить часть текста и перетащить на окно выбора стилей. После этого программа предложит сохранить текущий стиль как новый. Таким элегантным способом вы можете легко со-

здавать собственные оригинальные стили. Также применение стилей теперь допускается назначать на горячие клавиши.

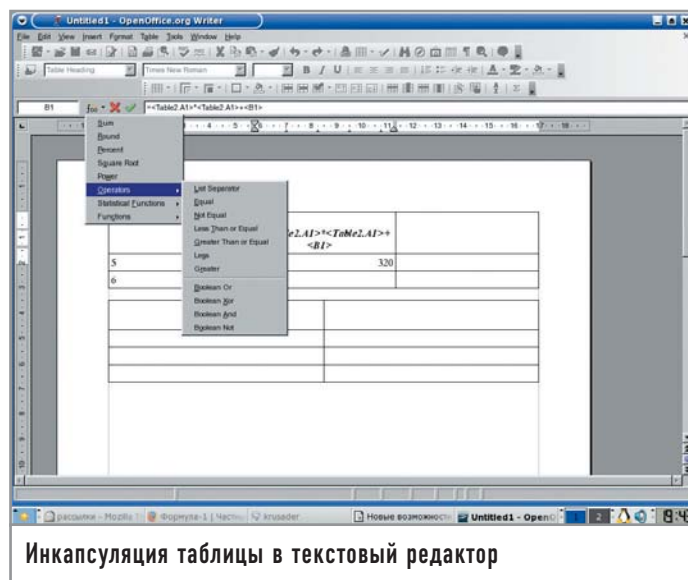
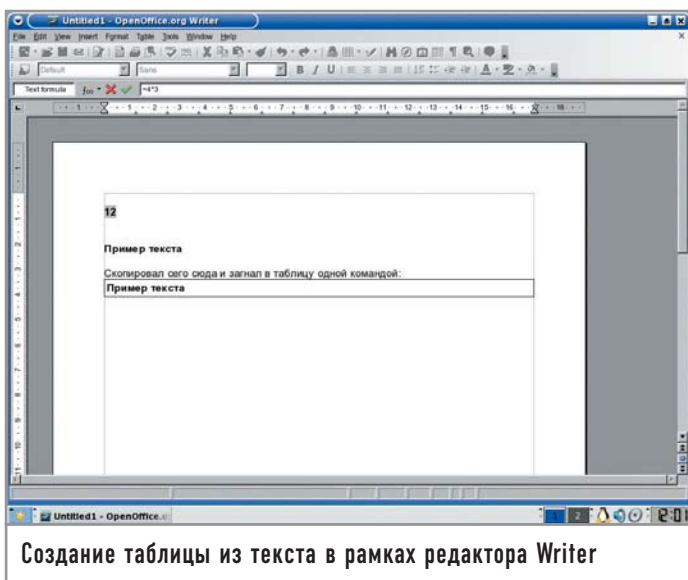
История последних открытых документов теперь по умолчанию хранит 10 записей, а в заголовке окна всегда пишется реальное имя файла, а не строка «Title» из свойств документа.

OpenOffice.org Writer

Самым востребованным приложением в офисном мире всегда был и остается текстовый редактор. Более половины документооборота (служебные записки, приказы, письма) осуществляется с его помощью, единственным же «конкурентом» можно считать электронные таблицы (накладные, авансовые отчеты, штатные расписания, сметы).

В меню данной программы также произошли своеобразные изменения — появился отдельный пункт «Table», отвечающий за работу таблиц, и теперь она стала поистине впечатляющей. Но основным новшеством второй версии можно смело назвать возможность создавать таблицы внутри уже существующих, причем с неограниченным уровнем вложения. Кнопка создания таблицы находится прямо на панели. Пользователю больше не нужно тянуться в левый край экрана, нажимать на кнопку «Вставить» и лишь затем выбирать пункт «Таблица». Еще одна полезная функция — это возможность создания таблиц непосредственно из текста. Для этого нужно лишь выделить текст и выбрать в меню функцию «Text to table», и текст окажется внутри автоматически созданной таблицы. Возможна и обратная функция. Вы выделяете в существующей таблице произвольное количество текста, затем выбираете пункт «Table to text», и этот текст форматируется в чистом виде. Ячейки можно отделять символами табуляции, абзацами, а также независимыми колонками — как вам больше нравится.

Под барабанную дробь следует произносить такое улучшение OpenOffice.org Writer как возможность вычисления в таблицах. После нажатия клавиши «F2» появляется строка, как в табличном процессоре, где можно назначать любые, даже самые замысловатые формулы. Вычисления могут производиться между соседними таблицами в пределах одного документа (файла).



Форматирование таблиц также стало выполнять намного проще. Появились новые опции и кнопки на панели для выравнивания, а существующие еще с давних времен теперь не запрятаны в глубоких подменю, а вызываются прямо с панели инструментов, из «Table» или контекстного меню. Есть также отдельные кнопки для выделения строк и столбцов таблиц, заливки и отдельных ячеек текстурами. В таблицах теперь можно оперировать сразу несколькими ячейками (такая возможность пригодится при редактировании больших документов). Направление текста указывается для каждой ячейки отдельно и по умолчанию совпадает с тем, что присутствует на странице.

Расширена статистика документа. Она теперь может показывать общее количество слов, символов и т. д., в том числе и для выделенного текста. Наконец-то на панели появилась кнопка предварительного просмотра страницы. Почему это не было сделано до сих пор — не понятно. Специальная вставка вызывается сочетанием клавиш «Ctrl+Shift+V».

С помощью комбинации «Ctrl+стрелка вверх» и «Ctrl+стрелка вниз» можно двигать параграфы в документе.

В документах, открытых только для чтения, доступно копирование информации в буфер обмена.

OpenOffice.org Calc

В программе OpenOffice.org Calc появилась возможность вставлять таблицы из внешних файлов через пункт «Insert → Sheet from file». Таблице может содержать до 65 536 строк вместо 32 000. Это сделано для совместимости с Microsoft Excel. Ячейки таблиц теперь могут иметь диагональные бордюры. Эта функция вызывается из меню «Format → Cells → Borders». Теперь в левом окошке можно выбрать диагональные линии. Тот, кто когда-нибудь составлял турнирные таблицы, оценит эту возможность: диагональное форматирование образует полосу, которая простирается от левого верхнего до правого нижнего угла. Ведь никто же не будет играть сам с собой — поэтому ячейки зачеркивают одной диагональной чертой.

Когда ячейки отфильтрованы с помощью AutoFilter, Standard Filter или Advanced Filter, ниже перечисленные дей-

ствия не сказываются на отфильтрованных рядах, действуя как выделение нескольких ячеек без этих рядов:

- ▶ удаление содержимого ячеек;
- ▶ удаление рядов (удаление отдельных ячеек выключено);
- ▶ изменение формата ячеек, включая стили.

Кардинальные изменения произошли в работе с примечаниями. В новом Calc открылись богатые возможности форматирования примечаний:

- ▶ окно можно двигать, менять размер;
- ▶ текст можно произвольно форматировать, применять стандартные атрибуты;
- ▶ стрелку и линию можно также гибко настраивать;
- ▶ можно менять цвет фона, делать его градиентным или заливкой текстурой.

Текст в таблицах можно теперь писать справа налево, что делает возможным использование программы для составления документов на азиатских языках.

OpenOffice.org Impress

У программы полностью сменился движок, что позволяет использовать в презентациях множество ранее недоступных эффектов. Для повышения совместимости между Impress и Microsoft PowerPoint расстояние между строками больше не зависит от используемого шрифта.

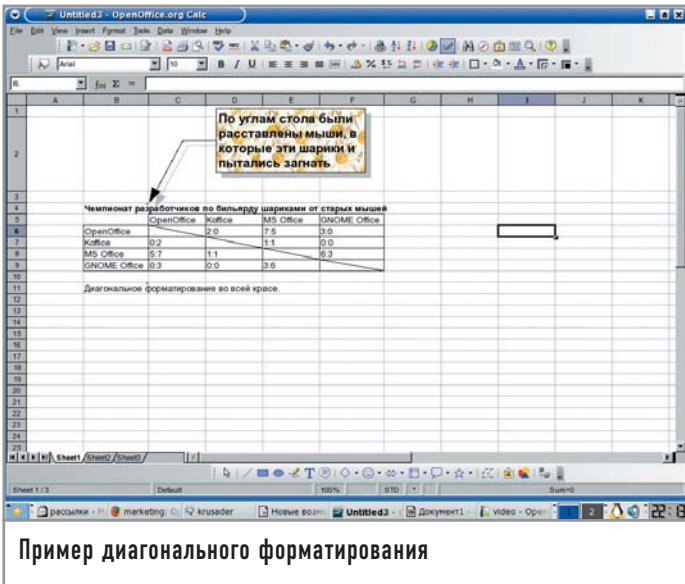
OpenOffice.org Draw

Панель инструментов рисования полностью переделана, для того чтобы обеспечить работу с новыми видами фигур: «Basic», «Block Arrow», «Symbol», «Flowchart» и «Stars & Banners».

Геометрию фигур в «Custom Shapes» можно гибко редактировать и преобразовывать в трехмерный формат, а также использовать в утилите Fontwork для рисования потрясающе красивых надписей (говоря языком GIMP — эмблем).

OpenOffice.org Base

В этом новом отдельном приложении вы можете создавать формы, рапорты, таблицы. Все базы данных хранятся в одном файле с общим форматом данных. Драйверы OpenOffice.org



2.0 позволяют запускать цифровые, строчные, а также функции «дата-время», описанные в SQL92 для dBase, flatfile и электронных таблиц.

Для создания новых баз данных в OpenOffice.org Base создана мощная система различных Wizard, что позволит даже далеким от тонкостей работы с базами данных людям быстро освоиться в новом приложении.

Итоги

Время показывает, что единственным серьезным конкурентом Microsoft Office в течение последних лет является только OpenOffice.org. В первую очередь это обусловлено тем, что в OpenOffice.org уже давно имеется приемлемая поддержка форматов MSO, а значит возможна безболезненная миграция пользователей. Сегодня, хотя это и юмор, можно сказать, что MS Office тоже стал кросс-платформенным приложением — настолько здорово он заводится в Linux из-под CrossOver Office 4.0! Но если серьезно, реальным плюсом OpenOffice.org является то, что на фоне роста популярности Linux этот офисный пакет стал мостиком, связывающим две абсолютно разные операционные системы. На рынке программ для работы в Интернете таким связующим звеном являются, пожалуй, продукты Mozilla.org.

Налицо тенденция к упрощению делопроизводства, многие новые функции ориентированы на облегчение трудоемких операций и оказание помощи в освоении продукта новыми пользователями. В очередной раз разработчики уделили много внимания совместимости. OpenOffice.org 1.0 обзавелся поддержкой MSO 97 и выше, в OpenOffice.org 1.1 эта поддержка улучшилась, к тому же добавился экспорт в PDF, а версия 2.0 несет очередное улучшение совместимости с MSO, экспорта в PDF, а также поддержку нового, утвержденного Европейской комиссией, стандарта OpenDocument.

	Windows	Linux
OpenOffice.org 1.1.4	58 Мбайт	76 Мбайт
OpenOffice.org 2.0	79 Мбайт	105 Мбайт

Команда разработчиков OpenOffice.org состоит всего из 50 человек, а как много они сделали и продолжают делать. Всего два года труда было потрачено на разработку новой версии, а продукт словно перешел из 9 класса школы сразу в аспирантуру. Гигантская фирма Microsoft с трудом поспекает за небольшой группой людей, с честью выполняющих свою работу. Нет, нельзя сказать, что OpenOffice.org опережает Microsoft Office в развитии, но факт максимального сближения их функциональных возможностей, несомненно, очевиден. Наше мнение такое, что лентяям и консервативным людям лучше работать с Microsoft Office. Пока еще рано с кровью отрывать людей от насиженных мест — благодарности можно и не дожидаться. Но наличие множества уникальных функций, поддержка OpenDocument, бесплатность, кросс-платформенность, истинная интеграция компонентов — все это крутится перед носом чертовски сладким пряником. Из-за абсолютно незащищенной реализации макросов документы MSO являются сегодня основным рассадником вирусов на предприятиях. А зачем вам это надо? Linux — лучший антивирус, это понятно, но есть менее радикальное антивирусное средство — OpenOffice.org, тем более что одно другому не мешает.

Если же вы уже давно работаете в Linux, смеем предположить, что OpenOffice.org вы видели, знаете и как минимум используете наряду с KOffice или GNOME Office. Тогда добро пожаловать в новый формат OpenDocument, заряженный полнофункциональным офисным пакетом для избранных, видящих мир в стиле Матрицы, и домохозяйек, роняющих слезу счастья при виде симпатичных Wizard. |

История создания OpenOffice.org

Дорогой успеха

OpenOffice.org был зачат в недрах немецкой фирмы Star Division, назывался в ту пору Star Office и носил статус free-ware. Когда пакет достиг версии 5.1, его популярностью заинтересовалась неизвестная фирма Sun, разработчик языка программирования Java. Sun купила Star Division и начала сама распространять Star Office 5.2, но уже на платной основе. В то же время Sun открыла его исходные тексты и основала проект OpenOffice.org — бесплатный офис на основе Star Office 5.2. OpenOffice.org стал быстро развиваться, меньше чем за год обзавелся полной поддержкой документов MS Office 97 и выше (до этого была поддержка только MS Office 95), был переведен на

многие языки мира, избавился от неудачного десктопа, в котором присутствовала своя панель задач и аналог кнопки «Пуск». Русскоязычные пользователи почувствовали себя в своей тарелке, когда за дело взялась фирма ALT Linux, которая не только перевела интерфейс офисного пакета, но и разработала русские словари для проверки орфографии и расстановки переносов. В 2002 году вышел Star Office 6.0, который отличался большим количеством приложений и широким набором вспомогательных инструментов. В 2003 году вышел Star Office 7.0, а OpenOffice.org 1.1 был признан лучшим офисным пакетом года на Open Choice Awards 2003.

На ринге легковесы

Одной из причин, по которой пользователи переходят с Windows на Linux, является возможность выбора графической оболочки. Мощные и многофункциональные KDE и GNOME, безусловно, хороши, однако многих не устраивает их большой размер и невысокая скорость работы. Но мир свободного ПО не ограничивается этими двумя графическими средами. В этой статье речь пойдет как раз об альтернативах KDE и GNOME.

| Xfce |

Xfce открывает наш обзор не случайно, так как не относится к оконным менеджерам типа Window Maker и Fluxbox — это именно окружение рабочего стола, как KDE и GNOME. Не бойтесь, скачивать монстра объемом 300 Гбайт не придется, Xfce с плагинами занимает около 18–23 Мбайт в архиве. Исходники и скомпилированные пакеты (кому как удобно) находятся на официальном сайте проекта: www.xfce.org. Последняя версия программы на момент написания статьи была 4.2.2. Прежде чем приступать к инсталляции Xfce, стоит убедиться в наличии необходимых библиотек и модулей в системе. Вот что нам понадобится: `pkgconfig`, библиотека GTK версии 2.2 и старше, `libxml2` и `libdb`. Также необходимы `libsvg` и `libstartup-notification`. Все это можно загрузить прямо с сайта разработчика.

Для установки менеджера в систему никакой ручной правки конфигурационных файлов не требуется, достаточно ввести в консоли `xfce_setup`, и при перезагрузке X-сервера вы увидите новый интерфейс. Вернуть старую конфигурацию можно командой `xfce_remove`.

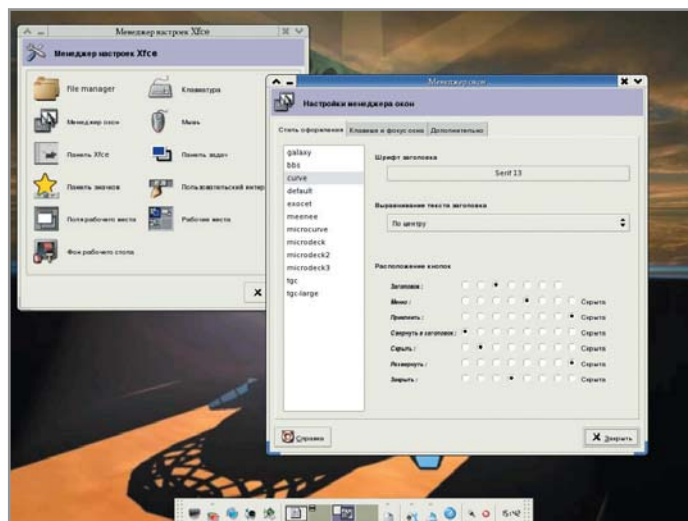
Первое, что бросается в глаза при старте Xfce, — это скорость загрузки. The Cholesterol Free Desktop Environment (именно так звучит полное название оконного менеджера) является производительным и нетребовательным к ресурсам окружением, не похожим ни на один другой диспетчер

окон для *nix. При первом запуске пользователя встретит девственно чистый рабочий стол с запущенным на нем переключателем виртуальных экранов и Xfce main panel — главной панелью, расположенной по центру экрана. После клика левой кнопкой мыши на рабочем столе появится меню, из которого можно будет запустить терминал, воспользоваться главным меню KDE или GNOME для запуска программ, настроить Xfce с помощью пункта «Settings», а также выключить или перезагрузить компьютер. Клик правой кнопкой мыши выведет список всех запущенных на текущем виртуальном экране программ. Управлять размерами окон можно посредством колесика мыши.

На главной панели находятся пункты меню, где сгруппированы различные утилиты. Например, в пункте «xterm» можно обнаружить несколько терминалов, в «Netscape» находятся браузеры и почтовые клиенты, а в «xedit» расположен набор текстовых редакторов. В любой пункт меню можно добавить свое приложение, выбрав из раскрывающегося диалога «add icon». Программы в Xfce можно запускать в определенной части экрана, что довольно удобно при работе с большим количеством приложений. Например, набрав команду `GIMP -geometry +50+50`, вы увидите растровый редактор посередине экрана. А команда `Gkrellm -xrm "*"Desk:3"` запустит Gkrellm на третьем виртуальном рабочем столе.



Мультимедиа в Xfce: XMMS и Mplayer



Настройка менеджера окон Xfce

Ознакомившись в общих чертах с оконным менеджером, будет благоразумно заняться его настройкой.

| Файловый менеджер Xffm |

Одной из главных изюминок Xfce является собственный файловый менеджер — Xffm, поэтому о нем следует рассказать отдельно. Щелкнув в его окне по выделенному документу правой кнопкой, вы увидите объемное меню пунктов из 15-17. Оно позволит вам открыть файл в новом окне, создать, удалить, переименовать файлы и каталоги, просмотреть свойства сфокусированного элемента, найти документы и многое другое. Копировать файлы удобнее всего перетаскиванием объекта из одного окна Xffm в другое (количество открытых окон не ограничено). Тут же можно найти утилиту для запуска программ, вызывающую панель для ввода пути к приложению. Если вы хотите, чтобы файловый менеджер при старте начинал, допустим, с директории /tmp/src, введите в консоли команду:

```
xffm /tmp/src
```

В общем, с Xffm разобраться несложно. Менеджер радует высокой скоростью, отличной функциональностью, мощной системой

поиска и удобством в работе. Могильщиком Midnight Commander он, конечно, не станет, но своих пользователей найдет.

| Графическая настройка Xfce |

Вся настройка Xfce может быть осуществлена через графический интерфейс, что является существенным плюсом для многих пользователей. Для запуска программы-конфигуратора надо кликнуть по кнопке «Settings» на главной панели. Перед вами откроется окно с несколькими пунктами. Начнем с файлового менеджера. Программа для настройки Xffm представляет собой окно, разделенное на две части — «Параметр» и «Значение».

Напротив «TERM» пропишите терминал, который используется в вашей системе, например `gnome-terminal`. Параметр «XFFM_HOME» нужен для определения вашей домашней директории, путь к которой и требуется ввести.

В правом нижнем углу файлового менеджера присутствует строка состояния, где выводится информация о директории, в которой находится пользователь, выделенном файле и состоянии корзины. Когда эта информация не помещается в строку, менеджеру приходится увеличивать размеры окна, что довольно неудобно.

Для «XFFM_STATUS_LINE_LENGTH» установите значение 48. Теперь на экран будут выводиться лишь первые 48 символов сообщения, поэтому деформации окна не произойдет. «XFFM_MAX_PREVIEW_SIZE» устанавливает максимальный размер картинок, изображения которых будут демонстрироваться прямо в окне файлового менеджера. Если изображение превышает допустимый размер, то оно показано не будет.

«SMB_USER» — здесь введите имя пользователя и пароль Samba в таком формате: `user%password` (если, конечно, пользуетесь этой программой). Напротив «SMB_CODESET» нужно ввести кодировку, используемую для отображения ресурсов компьютеров вашей локальной сети, например: `ISO-8859-1`.

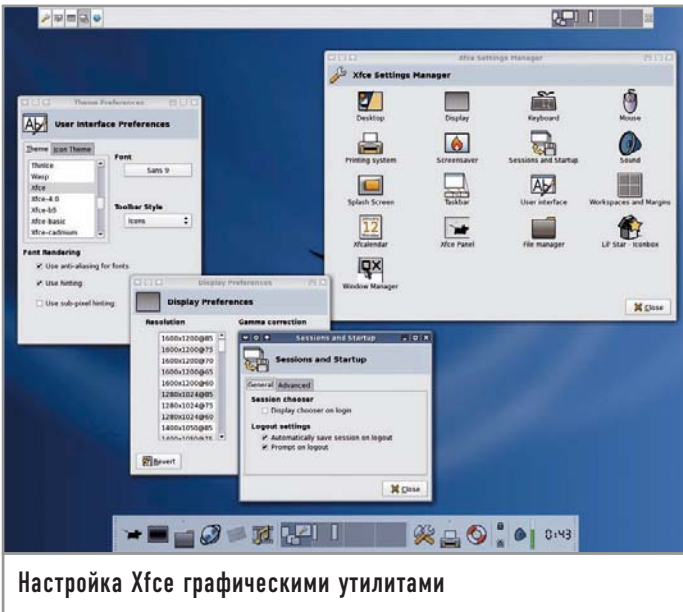
Если активировать опцию «XFFM_APPEND_FILES» значением «ON», файлы в папках будут располагаться по дате последнего изменения.

Параметры «XFFM_HIDE_BOOK», «XFFM_HIDE_LOCAL», «XFFM_HIDE_NETWORK», «XFFM_HIDE_APPS», «XFFM_

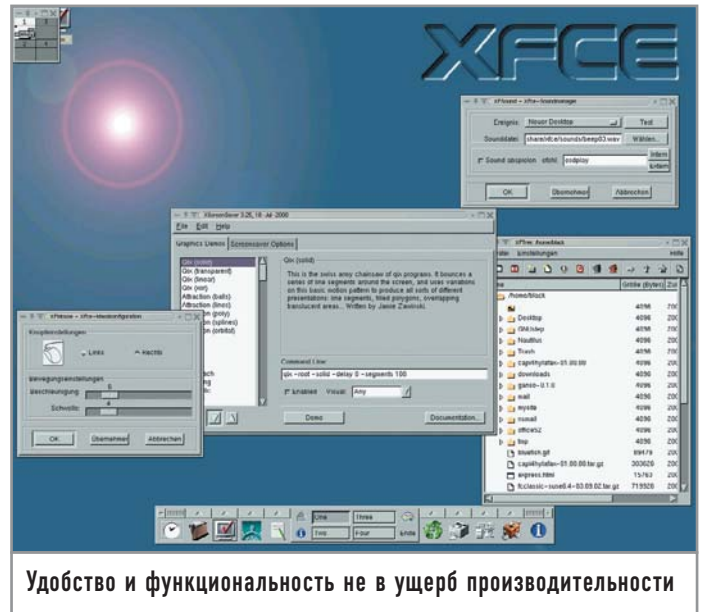
Продуктивная работа

Горячие клавиши Xffm

- «Alt+A» — выбор всех файлов в данном каталоге
- «Alt+Shift+A» — отмена выбора всего
- «Alt+D» — создание копии выделенного файла
- «Alt+Shift+D» — убрать/показать скрытые файлы
- «Alt+E» — очистка корзины
- «Alt+F» — поиск в данном каталоге
- «Alt+H» — перемещение в домашнюю директорию
- «Alt+I» — информация о выделенном файле
- «Alt+K» — создание нового файла
- «Alt+N» — создание новой директории
- «Alt+Q» — выход из программы
- «Alt+R» — переименование выделенного файла
- «Alt+W» — закрытие текущего окна
- «Ctrl+X» — удаление выбранных файлов



Настройка Xfce графическими утилитами



Удобство и функциональность не в ущерб производительности

HIDE_FIND», «XFFM_HIDE_TRASH», «XFFM_HIDE_FSTAB» можно включить, установив значение «ON». По умолчанию они выключены. Активированные опции будут удалены из главного меню соответствующими пунктами:

- ▶ «Book» — в этом меню следуют поместить наиболее часто используемые программы, что-то вроде панели быстрого запуска;
- ▶ «Local» — пункт для перемещения по дереву каталогов, по умолчанию начинается с домашней директории;
- ▶ «Network» — здесь находится все, что связано с Samba и локальными сетями;
- ▶ «Apps» — приложения, расположенные на главной панели; здесь можно создавать меню, выносить новые ярлыки программ на панель;
- ▶ «Find» — программа для поиска;
- ▶ «Trash» — корзина;
- ▶ «Fstab» — утилита для работы со сменными носителями, автоматически монтирующая CD-ROM и флоппи-дискетов.

После настройки файлового менеджера можно заняться клавиатурой. В настройках ввода активируйте опцию «Повторять». Благодаря ей можно не нажимать несколько раз на клавишу, а просто удерживать ее. Задержку и скорость повтора стоит делать короткими, иначе возникнут определенные неудобства. Можно установить мигание курсора, а также его скорость. Ну а для фанатов Emacs разработчики даже предусмотрели соответствующую схему клавиатуры с командами и горячими клавишами этого текстового редактора.

Наиболее важным с точки зрения удобства является пункт «Менеджер окон». Здесь можно выбрать стиль оформления окон и шрифт, выровнять текст заголовка. Кнопки «Закрывать», «Свернуть», «Развернуть» допустимо расположить хоть в центре, хоть справа панели заголовка, причем каждую кнопку можно поместить отдельно от остальных. Правда, лучше оставить параметры по умолчанию, иначе долго придется привыкать к новому расположению.

Продвинутые настройки Xfce

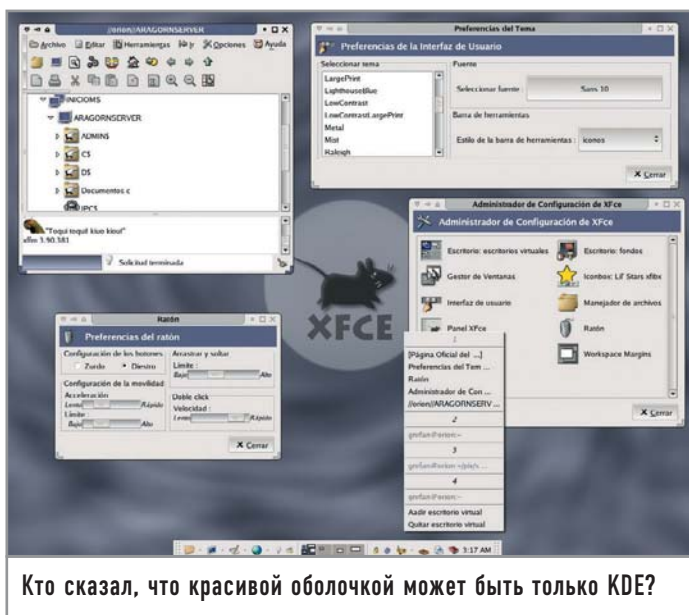
Редактирование конфигурационного файла

Графические средства конфигурирования, безусловно, хороши. Но всем настоящим юниксоидам уже давно известно, что наиболее гибко настроить программу можно лишь правкой файлов конфигурации. К тому же при грамотном использовании это на порядок быстрее. В Xfce разработчики использовали формат XML для создания файлов конфигурации. Наверное, это удобно для графических утилит конфигурирования, но вот для ручной правки не очень желательно. Усугубляет ситуацию то, что ни-

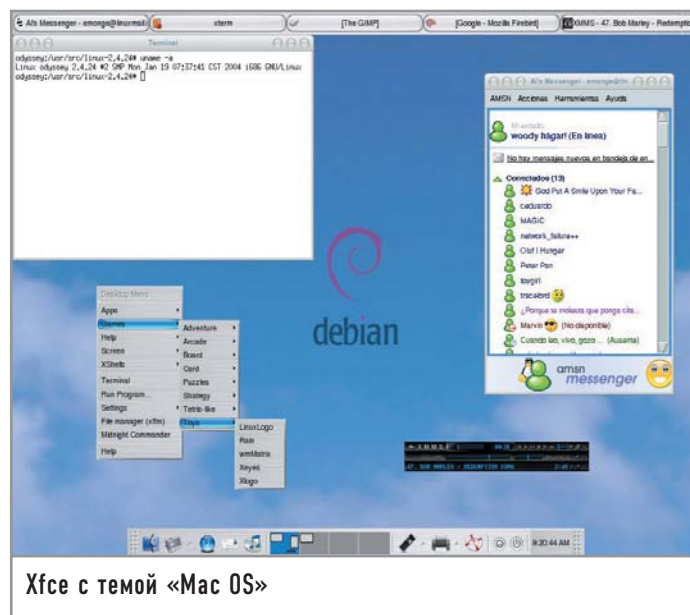
какого руководства и информации в мануалах о файлах настроек нет. Поэтому разбираться приходится интуитивно, анализируя синтаксис документа. Тем не менее некоторые файлы могут серьезно облегчить пользователю жизнь. Например, управлять главной панелью Xfce удобнее всего редактированием файла `~/xfce4/xfce4rc`. Пункты меню начинаются после тега `<Groups>`. `<Group>` //создаем новый пункт меню `<Control id="-2" popup="1">`

```
<Command term="0"
sn="1">xffm</Command>
//команда, которая будет
выполнена при клике по яр-
лыку этого пункта меню.
В данном случае — Xffm
<Tooltip> //файловые менед-
жеры
</Tooltip> //всплывающая
подсказка
</Control>
<Popup> //включаем в меню
дополнительные программы
<Item> //новая программа
<Caption>Konqueror</Caption>
//название
<Command term="0"
```

```
sn="0">konqueror</Command>
//команда запуска
<Tooltip>Konqueror File
Manager</Tooltip> //всплыва-
ющая подсказка
</Item> //здесь все аналогично
<Item>
<Caption>Nautilus</Caption>
<Command term="0"
sn="0">nautilus</Command>
<Tooltip>Nautilus File
Manager</Tooltip>
<Icon id="2"/>
</Item> //закрываем теги
обязательно для языка XML
</Popup>
</Group>
```



Кто сказал, что красивой оболочкой может быть только KDE?



Xfce с темой «Mac OS»

Окно получает фокус либо при щелчке по нему мышью, либо просто при наведении курсора. Есть возможность автоматически активировать вновь созданные окна, установить задержку перед поднятием активированного окна над другими.

Очень интересная функция — прилипание окон к краям экрана и другим окнам. Это позволяет удобно упорядочить программы на рабочем столе.

Есть опции, реализующие переключение между рабочими столами при достижении курсора мыши края экрана и перемещении окна приложения за экран. В настройках можно указать, будет ли при операциях перемещения и изменения размера демонстрироваться все окно или только его контур.

Четвертый пункт в менеджере настроек носит имя «Мышь». Нетрудно догадаться, какое устройство здесь конфигурируется. Можно установить схему кнопок для левой или правой, указать ускорение мыши и скорость двойного щелчка. Мы рекомендуем поставить большую скорость мыши при высоких разрешениях, и меньшую — при низких.

Теперь настроим пункт «Панель Xfce». Выберите размер панели, ориентацию (вертикальная/горизонтальная), тему знач-

ков, установите ее расположение. Кстати, главную панель Xfce можно поместить даже в центре экрана, не говоря уже о позициях сверху и по краю. Есть также возможность сделать панель автоматически скрываемой, она будет появляться лишь при наведении на нее курсора.

Еще один важный пункт в менеджере настроек — «Панель задач». Здесь можно установить расположение и размер панели задач, поместить на нее пейджер, виртуальные экраны и системный лоток. Интерфейс панели задач конфигурируется дополнительным пунктом — «Панель значков».

В разделе «Пользовательский интерфейс» можно выбрать тему оформления рабочего стола, палитру, шрифт, которые будут использоваться во всех приложениях Xfce.

Пункт «Поля рабочего места» определяет размер развернутого окна. Здесь надо установить отступы от края экрана в пикселях, за которые не будет выходить окно приложения.

В пункте «Рабочие области» введите число виртуальных рабочих столов и их названия.

Последний пункт в менеджере настроек — «Фон рабочего стола». В этом разделе выберите изображение для фона и различные эффекты к нему.

| Плагины Xfce |

Xfce имеет очень гибкую систему плагинов, что обеспечивает высокую функциональность этой среды. Рассмотрим плагины, которые были скачаны вместе с оконным менеджером, то есть самые распространенные. Всего их намного больше.

► Xfcllock — программа, включающая в себя часы, календарь, а также небольшой органайзер с ежедневником. Настройки лежат в файле `/home/user_name/.xfce/xfcllockrc`.

Нужно его немного подправить:

#использование нормальных шрифтов

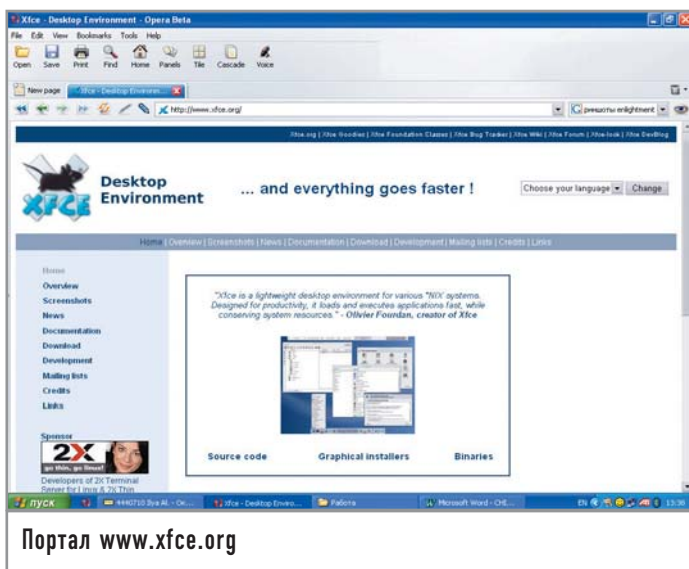
[Colors]

iso8859-15

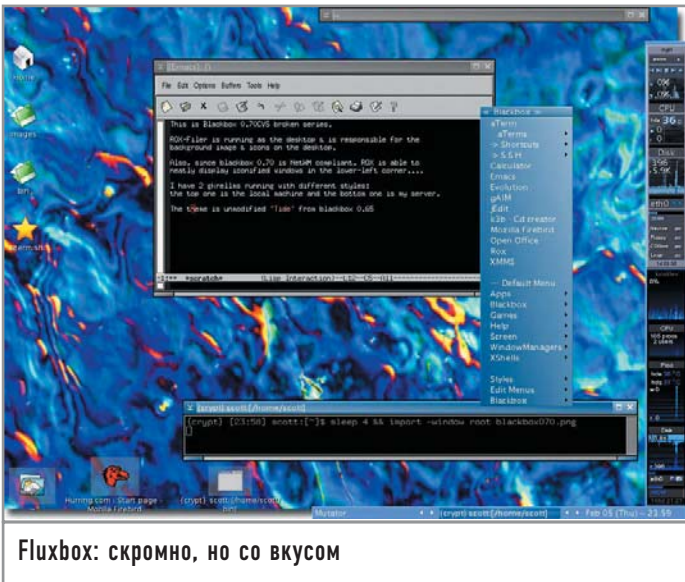
#отображение календаря и диалогов меню

DisplayCalendar

DisplayMenuBar



Портал www.xfce.org



Fluxbox: скромно, но со вкусом

#скрытие отображения плохо нарисованных часов

HideAnalogClock

#выделение праздников на календаре

DisplayHoly_days

#включение отображения секунд в часах

DisplaySeconds

► Xfbdmgr — небольшой плагин, позволяющий выбирать вид фона при запуске. Присутствует смена обоев через определенные промежутки времени.

► Xfglob — найдется все! Очень хорошая локальная система поиска, позволяющая искать по самым разным параметрам — начиная от расширений файлов и кончая датой изменения.

► Xfhelp — справку в Xfce должен прочесть каждый пользователь этой графической оболочки. Здесь представлено достаточно грамотное руководство.

► Xfmountdev — утилита для автоматического монтирования дискет и CD/DVD.

► Xfmouse — манипулятор ввода типа «мышь». Это важнейшее устройство при работе с компьютером. Стоит ли говорить, что оно должно работать хорошо? А с этим плагином вы можете с максимальным удобством настроить вашу мышь.

► Xfrpager — замечательный переключатель виртуальных рабочих столов. Учитывая то, что им пользуются все приверженцы Xfce, рассмотрим утилиту подробнее. Перед запуском добавьте в файл .xftmrc, который находится в домашней директории, следующее:

AddToFunc "InitFunction"

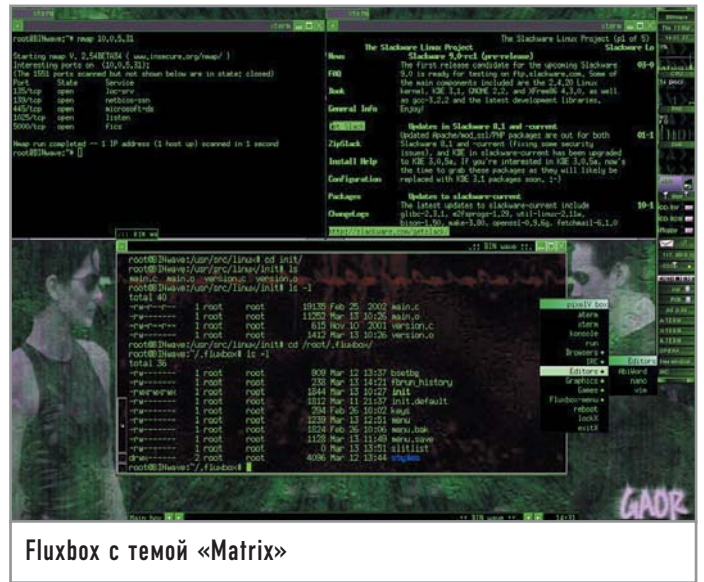
+ "I" Module xfrpager

Запускается программа командой xfrpager X, где X — число виртуальных рабочих столов. Для более удобной работы с Xfrpager нужно отредактировать вышеупомянутый .xftmrc. Так, чтобы разместить окна программы внизу, справа от главной панели, нужно добавить в него строку:

***xfrpager.geometry 63x5+637x616**

Xfrpager — панель, состоящая из окошек, представляющих собой уменьшенные копии виртуальных рабочих столов. Размер каждого окошка можно сделать равным 60x60 пикселей следующим образом:

***xfrpager.icongeometry 60x60**



Fluxbox с темой «Matrix»

Теперь пора подобрать шрифт для отображения названий рабочих столов:

***xfrpager.font iso8859-15**

Далее следует прописать название и количество рабочих пространств:

***xfrpager.label 1 Multimedia**

***xfrpager.label 2 Office**

***xfrpager.label 3 Internet**

► Xfplay — консольный проигрыватель аудио. Неплохой, но ведь уже есть mpg123.

► Xfprint — с переходом на Xfce не стоит отказываться от принтера, Xfprint поможет вам настроить печать (HP DeskJet 610c пятилетней давности распознавался без проблем).

► Xfsamba — этот плагин является клиентом Samba, которую создали, чтобы связываться с компьютерами, работающими под Windows.

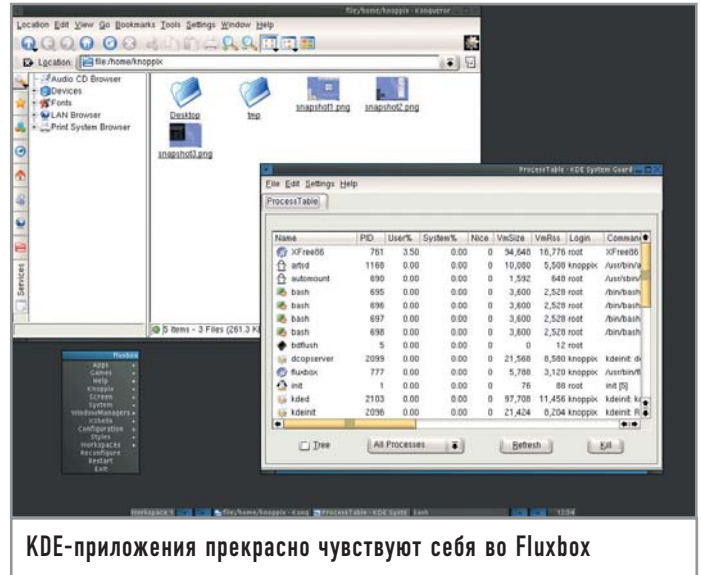
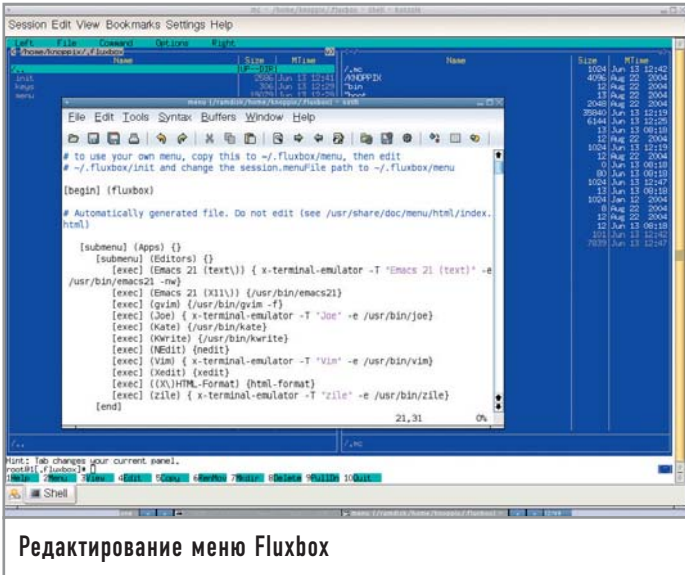
► Xfskin — красота спасет мир, а Xfskin украсит ваш десктоп интересными шрифтами, иконками и звуками.

► Xfsound — если у вас были проблемы со звуком, забудьте о них, с установкой этого плагина они благополучно исчезли.

► Xfbd — утилита для смены фона рабочего стола.



Поработав с Fluxbox, вы не сможете представить себе Linux без него



- Xfused — редактор меню.
- Xfce-battery — утилита для контроля уровня заряда батарей в ноутбуках.

Рассказать обо всех плагинах не представляется возможным в свете их огромного количества и ограниченного объема статьи. Предоставим читателю возможность самому ознакомиться с ними.

Fluxbox

О следующем оконном менеджере речь пойдет отдельно. Fluxbox является самой популярной из «легких» графических оболочек. Что же так привлекает пользователей во Fluxbox? Этот менеджер унаследовал только лучшее от своего предшественника — Blackbox, но при этом оброс дополнительными возможностями, такими как поддержка приложений KDE и GNOME, аплетов рабочего стола (dockapp) Window Maker, к тому же оконный менеджер обладает довольно приятным интерфейсом. Устанавливать Fluxbox лучше всего из исходников, которые можно скачать с сайта fluxbox.sourceforge.net, благо архив с программой весит меньше одного мегабайта. Так как вам нужна поддержка программ для дистрибутивов KDE и GNOME, скрипту `configure` при установке надо передать следующие параметры:

```
# ./configure --enable-kde --enable-gnome
```

После запуска Fluxbox вы увидите пустой рабочий стол и панель задач с часами и переключателем виртуальных рабочих столов внизу экрана. Несмотря на минимализм, оконный менеджер довольно красив, впечатление усиливается большим количеством тем. Дополнительные темы можно скачать на сайте themes.freashmeat.net.

Благодаря простому оформлению окон и отсутствию ненужных функций Fluxbox является одним из самых быстрых графических окружений. Еще одна интересная возможность — закладки, которые позволяют удобно разместить окна программ на рабочем пространстве. Несколько программ можно объединить в одну группу, например, если у вас запущено три терминала, их лучше сгруппировать и переключаться между ними по необходимости.

Графических утилит для настройки у Fluxbox практически нет — придется заниматься редактированием файлов конфигурации. Все настроечные файлы лежат в каталоге `/home/user_name/.fluxbox`.

Наверняка у каждого появится желание изменить меню и добавить туда свои программы. Для этого надо отредактировать `./fluxbox/menu`. Ниже приведена часть этого файла:

Вход в систему

Настройка KDM

Для загрузки графического интерфейса есть два пути: либо после входа в систему давать в консоли команду `startx`, которая запустит оконный менеджер, либо использовать X-клиент. Если у вас установлен только один оконный менеджер, лучше остановиться на первом варианте. Просто впишите в файл `~/xinitrc` строчку: `exec wm`, где вместо `wm` — имя вашего диспетчера окон. Если же вы используете несколько графических окружений или вам просто не хочется каждый раз вводить команду `startx`, лучше использовать графический клиент входа в систему. На сегодняшний день наиболее распространены три X-клиента: проект GNOME `gdm`, KDM для KDE и XDM от разработчиков X-Window. Мы же рассмотрим только KDM как функциональный и легкий в настройке. KDM можно установить вместе с KDE или скачать отдельно с

www.kde.org. Запустив терминал, повысьте свои привилегии до суперпользователя командой `SU` и запустите `kcontrol`. В разделе «System» выберете «Login manager», после чего откроется окно конфигурирования KDM. Нужно добавить различные оконные менеджеры в меню, чтобы при входе можно было выбрать, какой из них использовать. Для этого откройте пункт «Sessions» и кликните по кнопке «New type». В появившемся окне надо ввести имя диспетчера окон. Добавьте таким образом все графические среды вашей системы. После того как закончите, нажмите на кнопку «Apply» внизу окна. Нужно выполнить несколько команд:

```
rc-update add xfs default
rc-update add xfs default
```

Теперь осталось добавить в файл `/etc/rc.conf` строку `DISPLAYMANAGER="kdm"`, и можно приступать к работе.



Такая программа как Scribus загружается в Fluxbox мгновенно

[begin] (Fluxbox 0.9.13)

[exec] (xterm) {xterm -fg white -bg black -rightbar -fn 9x15 -geometry 100x40}

[exec] (kmail) {kmail}

[submenu] (Browsers)

[exec] (links) {links}

[exec] (mozilla) {mozilla}

[end]

[submenu] (Internet)

[exec] (ICQ) {centericq}

[exec] (IRC) {bitchx}

[end]

Файл начинается со слова begin в квадратных скобках и с номера версии Fluxbox — в круглых. Для того чтобы создать в меню пункт, вызывающий определенное приложение, его надо ввести ключевым словом [exec]. Далее в круглых скобках нужно прописать название программы, каким оно будет в меню, а в фигурных — команду, которая запустит приложение. Например, для вызова почтового клиента kmail нужно прописать пункт:

[exec] (kmail) {kmail}

Для создания подразделов существует ключевое слово [submenu], после которого в круглых скобках вводится имя подменю. После строки [submenu] (name_submenu) можно прописывать пункты запуска программ. Если помимо Fluxbox вы часто используете другой оконный менеджер, например KDE, добавьте строчку [restart] (Start KDE) {KDE}.

Теперь можно быстро переключаться между двумя графическими средами. Fluxbox поддерживает горячие клавиши. Различные комбинации можно прописать в ~/.fluxbox/keys. Например:

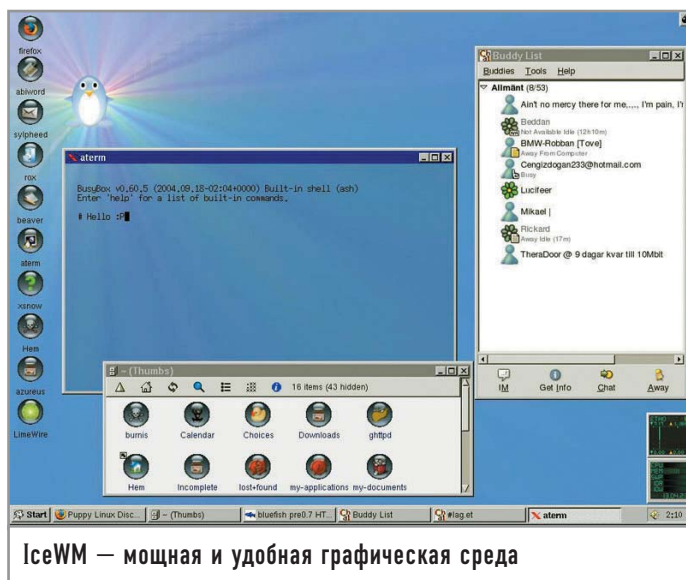
Mod1 F1 :Workspace1

Mod1 F2 :Workspace2

Mod1 P :ExecCommand xmms

Здесь Mod1 — это клавиша «Alt». Первые два сочетания клавиш нужны для перемещения между виртуальными рабочими столами, а третье запустит XMMS. Естественно, вместо проигрывателя музыки может выступить любое приложение.

Во Fluxbox расположение кнопок «Развернуть», «Свернуть», «Закрыть», находящихся в заголовке окна, можно поменять.



IceWM — мощная и удобная графическая среда

Для этого нужно отредактировать файл ~/.fluxbox/titlebar:

Right: sticky maximize

Left: close minimize

Как легко можно догадаться, теперь кнопки «Закрыть» и «Свернуть» располагаются слева в заголовке окна, а не справа, как обычно. Но наибольшее значение имеет файл ~/.fluxbox/init, являющийся главным файлом конфигурации. Здесь можно выбрать стили оконного менеджера. По умолчанию они хранятся в /usr/local/share/fluxbox/styles, но для удобства их лучше скопировать в ~/.fluxbox. Теперь нужно прописать пути к файлам настроек:

session.styleFile: /home/yumi/.fluxbox/styles/asket //путь к стилю

session.keyFile: /home/yumi/.fluxbox/keys //конфигурация горячих клавиш

session.titlebarFile: /home/yumi/.fluxbox/titlebar //конфигурация заголовка окна

session.menuFile: /home/yumi/.fluxbox/menu //конфигурация меню

Можно изменить имена рабочих столов в панели задач:

session.screen0.workspaceNames: one-net,two-work,three-multimedia

После «Names» вводите названия через запятую. Если необходимо, чтобы на панели помимо времени отображалась и дата, добавьте строчку:

session.screen0.strftimeFormat: %k: %M %p%d/%m/%y

Не нравятся закладки? Отключить их можно так:

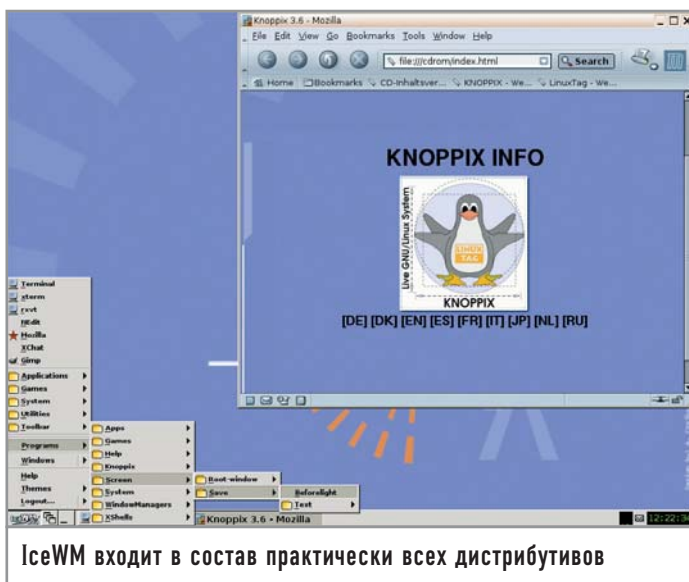
session.tabs: false

Для того чтобы вместе с оконным менеджером автоматически запускались некоторые программы, имеет смысл отредактировать ~/.xinitrc. Откройте этот файл текстовым редактором и добавьте сначала exec fluxbox, а затем все остальные приложения, которые должны запускаться вместе с оконным менеджером. Имя каждой программы должно быть на отдельной строчке, в конце которой присутствует знак амперсанда (&). Например:

exec fluxbox

xterm &

xmms &



IceWM входит в состав практически всех дистрибутивов

Есть и другие способы добавить программу в автозапуск, но этот является самым быстрым.

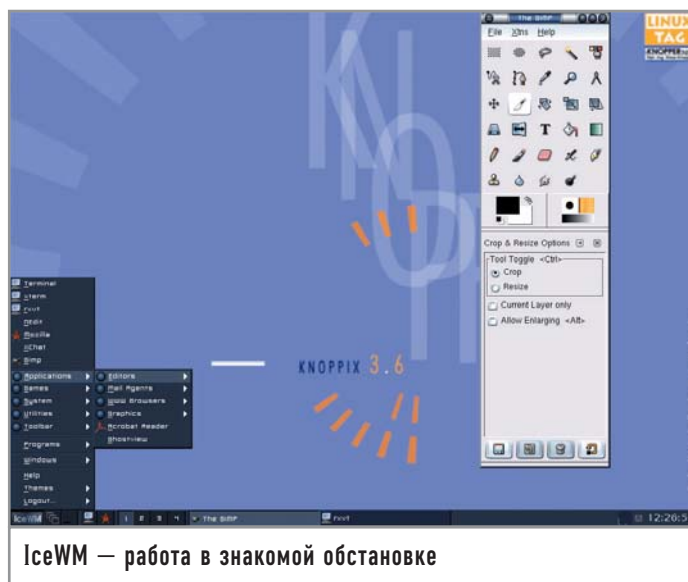
Для установки фона во Fluxbox удобно пользоваться утилитой fbsetbg, которая идет вместе с оконным менеджером по умолчанию. Для установки изображения в качестве полноэкрannого фона введите команду fbsetbg -f /tmp/image.jpeg. Если нужно, чтобы программа отображалась на всех рабочих пространствах, запускайте ее с параметром -w. (например:

Enlightment

Оконный менеджер для эстетов

Еще один оконный менеджер, о котором хотелось бы рассказать, — Enlightenment. Скачать его можно на сайте enlightenment.org. Первое, что бросается в глаза, — потрясающая красота WM. Он обладает огромным количеством классных тем, сложным оформлением окон, поддерживает сглаживание шрифтов TrueType, включает в себя море спецэффектов (анимация, звуки). Работает он на всех UNIX-системах, и не только: Linux, BSD, Solaris, HP, Irix, AIX. Разработчиками написаны библиотеки, используемые в работе другими менеджерами окон. Есть хорошая графическая программа конфигурации WM. В состав Enlightenment входит звуковой демон ESD, смешивающий аудиопотоки и позволяющий проигрывать звуки самого оконного менеджера, смотря при этом фильм или слушая музыку. Имеется переключа-

тель виртуальных столов, на котором изображены уменьшенные снимки того, что происходит на других рабочих пространствах. Проект имеет свой терминал Eterm, хорошо конфигурируемый и поддерживающий псевдопрозрачность. Enlightenment легко локализуется, не имеет проблем с русскими кодировками и шрифтами. Почему же эта среда так слабо распространена, если она такая хорошая? Главная причина в том, что для оконного менеджера Enlightenment слишком тяжел, к тому же довольно сильно тормозит. А люди переходят с GNOME и KDE во многом ради скорости. К тому же анимация, многочисленные эффекты — через пару часов работы все это начинает раздражать. Но если покопаться в настройках и удалить ненужные функции, WM вполне подойдет для постоянного использования.



IceWM — работа в знакомой обстановке

gkrellm -w). Что ж, теперь вы знаете достаточно для комфортной работы в оконном менеджере Fluxbox.

| IceWM |

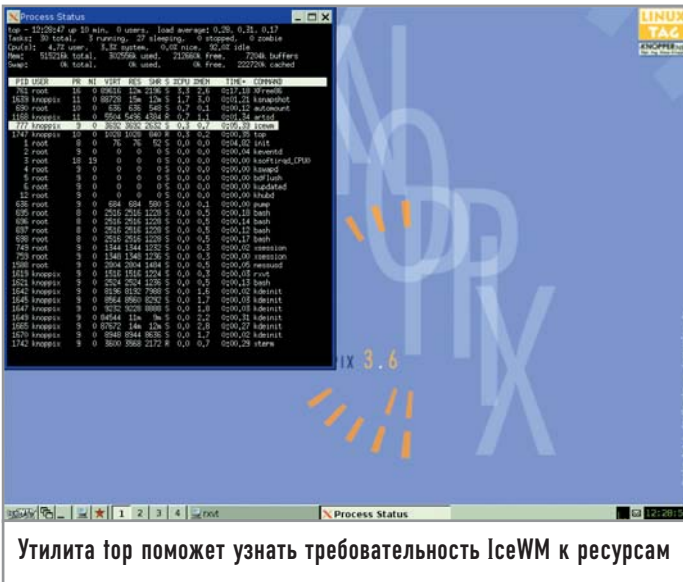
Графическая среда IceWM была разработана не так давно, но на сегодняшний день входит в состав практически всех известных дистрибутивов. Такая популярность обусловлена главным образом внешним видом диспетчера окон, который может имитировать стандартный интерфейс Windows, OS/2, Mac OS. Знакомая панель задач, иконки приложений для быстрого запуска, меню «Пуск» — все это делает IceWM лучшим выбором для новичков. Впрочем, производительность и мощность в конфигурировании оценят по достоинству и профессионалы UNIX. Минимальные системные требования — Pentium 166 МГц и 16 Мбайт оперативной памяти, что наверняка обрадует владельцев старых компьютеров. Официальный сайт IceWM — icewm.sourceforge.net, там можно найти дистрибутив программы и различную документацию.

При компиляции из исходных текстов надо добавить несколько опций конфигурирования:

```
#./configure --with-imlib=/usr/bin --with-gnome-menus=/usr/bin --enable-shaped-decorations --enable-gradients --enable-antialiasing --enable-xfreetype
```

Опция --with-imlib=/usr/bin позволяет использовать в качестве иконок рисунки любых форматов, нужно только указать месторасположение файла imlib-config.

Благодаря ключу --with-gnome-menus=/usr/bin IceWM импортирует в себя меню GNOME. К сожалению, с KDE такое повернуть не удастся, только с GNOME. Остальные опции нужны для придания менеджеру более симпатичного вида. Настраивается IceWM правкой конфигурационных файлов, лежащих в каталоге /usr/local/lib/X11/icewm, который для удобства можно скопировать в домашнюю директорию. Всего файлов пять: menu, toolbar, winoptions, keys и preferences. Начнем с редактирования меню, за которое отвечает файл menu. Для ввода пункта, запускающего программу, надо прописать ключевое слово «prog», затем ввести название приложения, каким оно будет в меню, путь к иконке и исполняемому файлу. Например:



Утилита top поможет узнать требовательность IceWM к ресурсам

```
prog "Mplayer" /usr/share/icons/mini/mplayer.xpm
/usr/bin/mplayer
```

Для создания подраздела в меню используйте параметр menu, после которого в фигурных скобках вводите нужные программы:

```
menu "Multimedia" /usr/share/icons/mini/multimedia.xpm
{prog "mp3blaster" /usr/share/icons/mini/mp3.xpm
/usr/bin/mp3blaster
prog "XawTV" /usr/share/icons/mini/xawtv.xpm /usr/bin/xawtv
prog "Mplayer" /usr/share/icons/mini/mplayer.xpm
/usr/bin/mplayer
prog "XMMS" /usr/share/icons/mini/xmms.xpm
/usr/local/bin/xmms
}
```

Чтобы добавить новую программу на панель быстрого запуска, надо изменить содержимое /usr/local/lib/X11/icewm/toolbar. Синтаксис аналогичен файлу menu:

```
prog "SIM" /usr/share/icons/mini/sim.xpm /usr/bin/sim
```

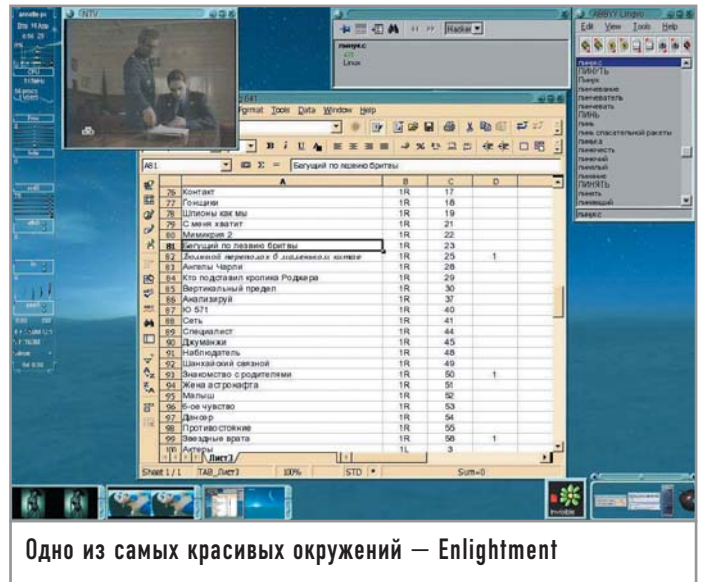
IceWM поддерживает горячие клавиши. Для установки новой комбинации надо подправить файл keys. Формат такой: key «сочетание+клавиш» команда. Например, чтобы запустить xterm одновременным нажатием клавиш «Alt» и «Shift», нужно прописать:

```
key "Alt+shift" xterm
```

Наибольшее количество настроек находится в файле preferences. Тут можно настроить шрифты, внешний вид, поддержку мультимедийных кнопок клавиатуры и мыши, количество виртуальных рабочих столов и многое другое — всего около 180 пунктов.

Файл представляет собой список параметров, которые имеют либо значение 1 (разрешены), либо 2 (запрещены). Ниже приведена часть файла preferences с комментариями.

```
QuickSwitch = 1 //разрешить переключение между окнами по
«Alt+Tab»
ManualPlacement = 0 //отмена ручного размещения окна на
рабочем столе
TaskBarAtTop = 1 //расположение панели задач сверху экрана
TaskBarAutoHide = 0 //отключение автоматического скрывания
панели задач
TaskBarShowWorkspaces = 1 //показать на панели задач пере-
```



Одно из самых красивых окружений — Enlightenment

ключатель между виртуальными рабочими пространствами

```
TaskBarShowClock = 1 //показывать часы
```

```
TaskBarShowCPUStatus = 0 //отмена показа состояния
```

```
загрузки процессора на панели задач
```

```
UseMouseWheel = 1 //поддержка колесика мыши
```

```
MultiClickTime = 300 //время в миллисекундах, требуемое
для распознавания двойного клика мыши
```

```
ClockFontName = "iso8859-15" //шрифт для отображения
часов
```

```
DesktopBackgroundImage = "/home/image/fon.xpm" //рисунок
для фона рабочего стола
```

```
WorkspaceNames= "NET", "Multimedia", "games" //названия
рабочих столов
```

```
Theme = "/usr/local/X11/icewm/default.theme" //путь
к используемой теме
```

Все параметры перечислить невозможно, поэтому лучше всего ознакомится с man-страницами.

Файл winoptions нужен для настройки окон конкретных приложений. В IceWM каждое окно программы имеет свой класс и свое имя. Эти значения можно узнать с помощью утилиты xprop, которую вы найдете на официальном сайте IceWM. Строки в файле winoptions должны быть такого вида:

```
класс_окна.имя_окна.параметр: значение
```

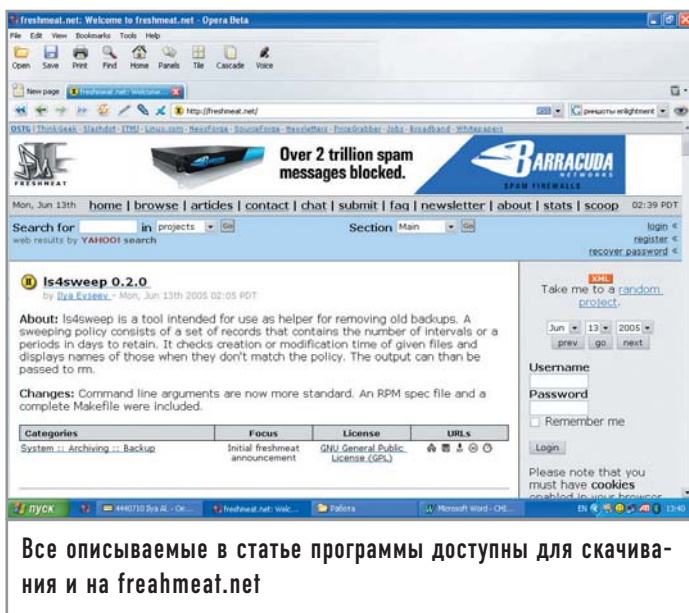
Как определить имя и класс, нам уже известно. Вот некоторые параметры окна:

- ▶ Icon — иконки приложения;
- ▶ Workspace — рабочий стол, где должно быть запущено окно программы;
- ▶ Geometry — местоположение окна на рабочем столе;
- ▶ Dborder — отключение рамки у окна;
- ▶ IgnoreTaskBar — окно не появится на панели задач.

Например:

```
Navigator.Netscape.workspace: 1
```

Для IceWM существуют специальные утилиты конфигурирования. С сайта icewm.sourceforge.net скачайте утилиту icepref. Она написана на Python и требует наличия в системе интерпретатора. Еще одна программа, призванная облегчить жизнь пользователям IceWM, — IceWMConf. Она доступна на сайте



icewm.tux.nu/tools/icewm-conf/Download. Обе утилиты не идеальны, но помогут избавиться от большого количества ручной работы типа составления меню и выбора стиля.

IceWM — очень хороший менеджер окон, оставивший приятное впечатление. Его можно порекомендовать пользователям, требовательным к производительности, но не желающим расставаться с функциональностью.

| Иконки на рабочем столе |

Если раньше вы работали с KDE или GNOME, то наверняка не можете представить себе рабочий стол без ярлыков программ. Но описываемые в этой статье оконные менеджеры не поддерживают иконки на рабочем столе. Так что же делать? Нужно использовать специальное ПО. Лучшая программа такого рода — IDesk, которую можно скачать с сайта idesk.sourceforge.net. После установки программы создайте в домашней директории файл `.idescrc`, в котором пропишите следующее:

table Config

```
FontName: verdana
FontSize: 12
FontColor: #ffffff
Locked: false
Transparency: 100
Shadow: true
ShadowColor: #000000
ShadowX: 1
ShadowY: 2
Bold: false
ClickDelay: 300
IconSnap: true
SnapWidth: 55
SnapHeight: 100
SnapOrigin: BottomLeft
SnapShadow: true
SnapShadowTrans: 200
CaptionOnHover: false
end
```

table Actions

```
Lock: control right doubleClk
Reload: middle doubleClk
Drag: left hold
EndDrag: left singleClk
Execute[0]: left doubleClk
Execute[1]: right doubleClk
end
```

Далее нужно создать директорию `~/ .idestop`. В ней будут находиться иконки и файлы конфигурации. Допустим, вам нужен ярлык программы GNUcash на рабочем столе. Сначала нужно создать файл `gncash.lnk` в директории `~/ .idestop`:

```
# touch /home/user_name/.idestop/gncash.lnk
```

Теперь в файл `gncash.lnk` добавьте следующие строчки:

table Icon

```
Caption: GNUcash //название программы
Command: gncash //команда, которая будет выполнена при клике по ярлыку
Icon: /home/user_name/.idestop/cash.png //путь к изображению, которое используется как иконка
end
```

Сделайте это со всеми программами, ярлыки которых вы хотели бы видеть на рабочем столе. Теперь нужно запустить IDesk: `nohup idesk > /dev/null &`

После этого в `.idescrc` вместо параметра `false` напротив `Locked` введите «true».

Осталось добавить IDesk в автозагрузку:

```
rc-update add idesk default
```

Теперь иконки опять присутствуют на вашем рабочем столе!

| Эпилог |

На этом мы заканчиваем обзор менеджеров окон для Linux. Новичкам, недавно пришедшим из мира Windows, следует обратить внимание на IceWM, интерфейс которого схож с внешним видом операционной системы от Microsoft. Аскетам, для которых главное — минимализм и скорость работы, стоит остановить выбор на Fluxbox. Ну а тем, кто предпочитает мощные и многофункциональные системы, сам бог велел ставить Xfce. В общем, выберите тот, который будет вам по душе. Удачи в работе с Linux! |

Дополнительная информация

Полезные ссылки

- <http://xfree86.org.ru> — официальный русскоязычный сайт графического сервера XFree86;
- <http://freshmeat.net/dockapp> — докаппы для Fluxbox и других менеджеров экранов;
- <http://bensinclair.com/dockapp> — еще один сайт с докаппами;
- <http://themes.org> — темы для всех диспетчеров окон;
- <http://themes.freshmeat.net> — самая большая коллекция тем в Интернете;
- <http://fluxbox.sourceforge.net> — официальный сайт Fluxbox;
- www.xfce.org — официальный сайт Xfce;
- <http://icewm.sourceforge.net> — официальный сайт IceWM;
- www.enlightment.org — официальный сайт enlightenment.

Linux?

Вам какой?

Критерием выбора дистрибутива для большинства пользователей становится обычно или программное наполнение, или система установки пакетов, или тот факт, что именно такая операционная система стоит у знакомого гуру. Но, в конце концов, решающим фактором при выборе системы становится ядро.

Что такое Linux? Внимательный читатель правильно (и наверняка с улыбкой) вспомнит, что на самом деле Linux — это ядро, и не более того. Важная, но все же лишь деталь в сложном механизме полноценной операционной системы. А уж дистрибутивы — это дело десятое. Нынешние пользователи дистрибутивов GNU/Linux редко задумываются о том, какое ядро установлено в их системе, а также что его можно и нужно обновлять, конфигурировать и компилировать. Это в целом хороший признак того, что Linux действительно честно отрабатывает слоган «Just works» в большинстве конфигураций, заботливо подобранных для пользователей разработчиками дистрибутивов. Однако не всех и не всегда удовлетворяют стандартные конфигурации ядер. Например, нынче стало модно компилировать ядра для PC-платформы вообще без поддержки шины ISA. Кто-то предъявляет повышенные требования к безопасности, и ему нужны самые последние исправления. Ну а кто-то хочет проникнуться тем самым духом fun, окружающим Linux, и опробовать какие-либо новшества, которых нет ни в одном дистрибутиве и которые доступны только тем, кто не боится скомпилировать ядро со своей (что на самом деле еще более страшно) конфигурацией. Поэтому начнем мы, собственно, с описания процесса сборки и сопутствующих ему действий.

| Официальное ядро Linux |

Во-первых, сразу же проясним, что значит «официальное ядро Linux». Сам Линус Торвалдс говорил, что официального ядра Linux не существует, есть лишь различные ветви, ведущиеся разными разработчиками; просто так сложилось, что ветвь основателя системы пользуется наибольшим доверием у пользователей. Что ж, это так, но именно его ветвь все-таки зачастую называется официальной, и именно от нее отталкиваются все параллельные. Официальное ядро Linux всегда доступно во всех разновидностях на <http://kernel.org>, но к этому мы еще вернемся, а сейчас перейдем к сборке.

| Сборка |

Итак, вы скачали около 40 Мбайт исходных кодов официального ядра актуальной версии. Что с ними делать? Нет никаких ограничений по тому, куда вы будете распаковывать ядро и где вы его будете конфигурировать и компилировать, но в целом его все-таки принято располагать в каталоге `/usr/src/linux-uname -r`. Принцип понятен, `uname -r` — команда, выдающая полную версию текущего ядра. Это значит, что ядро версии 2.6.12 будет логично поместить в каталог `/usr/src/linux-2.6.12`, поскольку многие сторонние скрипты-помощники ориентированы именно на такое расположение.

Теперь ядро необходимо сконфигурировать. Это делается командами `make config`, `make menuconfig` или `make xconfig`. Все они, по сути, идентичны, только представлены в разных обличиях. Первую команду вводить не рекомендуем, иначе через час-другой у вас покраснеют и остекленеют глаза, а окружающие подумают, что вы сходите с ума (проверено на личном опыте). Все эти неприятности могут случиться из-за того, что опций конфигурирования у ядра тьма тьмушая, а `make config` работает очень просто — последовательно требует у вас вразумительного ответа по каждой опции в виде «да/нет/модуль». Это долгий, требующий внимания процесс, поэтому лучше пользоваться `make menuconfig` и `make xconfig`. Первая программа состоит из текстовых меню, по которым можно гораздо проще ориентироваться и значительно быстрее сконфигурировать ядро, а последняя, соответственно, представляет собой полностью графический конфигуратор. По умолчанию он использует Qt, но можно набрать `make gconfig`, и конфигуратор будет собран для GTK. Полагаем, должно быть очевидно, что для этого потребуются заголовочные файлы Qt или GTK, впрочем, для `ncurses` (используемого в `menuconfig`) они тоже нужны (все конфигураторы собираются в системе при запуске).

Давать советы по конфигурации — дело неблагодарное, так что лучше будет найти информацию по этой теме в Сети и просто уяснить разницу между статической компиляцией, компиляцией модуля и некомпилированием модуля или функции вообще. Вся конфигурация ядра будет сохранена в один-единственный файл `config` в каталоге с ядром. Его полезно куда-нибудь скопировать, чтобы иметь впоследствии стартовую точку для любых дальнейших конфигураций: обычно его кладут в каталог `/boot` рядом с уже скомпилированным ядром в виде файла «`config-версия-ядра`». Кстати, позже, при обновлении ядра и наличии отлаженной конфигурации, вписать эту самую конфигурацию в новое ядро очень легко — копируете тот самый `config` и набираете «`make oldconfig`». Вас спросят только о новых опциях (если они будут), и вы будете готовы к компиляции.

Компиляция тривиальна — `make`, и все дела! В случае с веткой 2.4 чуть сложнее — `make bzImage` и `make modules`. Далее можно созерцать процесс компиляции. В нем красиво показывается сборка ядра, иногда некрасиво проскакивают многочисленные «warning» (в последних версиях их всего около 200 штук в конфигурации «yes» для всего, однако большей их части можно не придавать значения). По окончании этого процесса стоит установить модули ядра (если, конечно, они присутствуют в вашей конфигурации) по команде `make modules_install`. Все модули при этом аккуратно поместятся в директорию `/lib/modules/`. Наконец, следует установить само ядро с помощью `make install` или же скопировать его вместе с файлом `System.map` (его необходимо переименовать в «`System.map-версия-ядра`») вручную из директории `arch/i386/boot/bzImage` в `/boot`.

После этого конфигурируем загрузчик. Эта процедура зависит от того, что вы используете в вашей системе — GRUB или LILO, и ее мы описывать не будем (читайте — `man grub`, `man lilo`). Мы настоятельно рекомендуем GRUB, поскольку посто-

янно переустанавливать LILO просто-напросто неудобно, да и параметры командной строки у GRUB довольно мощные.

Хочется сказать пару слов о самих патчах. Как правило, это один файл, упакованный чем-нибудь наподобие `bzip2`, применяющийся с помощью команды `patch -p1` в каталоге исходников. Обычно легко сделать что-то подобное:

```
bzcat bla-bla.patch.bz2 | patch -p1
```

После такого некраткого ликбеза мы переходим к самому интересному — рассмотрению конкретных версий ядра Linux. Так уж получается, что этот обзор во многом является парадом разработчиков Linux, поскольку у каждой ветки есть главный разработчик, и он же, как правило, является основным, если и не кодером, то, по крайней мере, «приемщиком» патчей для своей ветки.

| 2.6.x |

Конечно же, логично начать с самого главного, а конкретно — с ветки Линуса Торвальдса. Мы не случайно говорили о том, что это всего лишь одна из веток ядра, и «официального» ядра Linux как такового не существует. Тем не менее все ориентируются на ветку Линуса, и она определенно главная в ядре, поэтому ее принято называть «официальной». Про нее, пожалуй, много не скажешь, кроме разве того, что ветка 2.6.x значительно отличается по модели разработки от 2.4.x — в ней допускаются значительно большие изменения без старта нестабильной ветки 2.7. Это способствует ее быстрому развитию и обновлению, а также позволяет производить полезные изменения, что очень хорошо, но в то же время, по общему признанию, ветка 2.6 в целом не такая стабильная, какой была 2.4. Конечно же, эти релизы можно найти на <http://kernel.org>

| 2.4.x |

Несмотря на то что ветка 2.4.x уже давно не является основной, она по-прежнему поддерживается. Более того, некоторые дистрибутивы (к примеру, Slackware 10.1 и многие Live CD) используют в качестве основы именно ее. Поэтому нелишним будет сказать, что эту ветку нынче ведет Марсело Тосатти, и в последнее время объем патчей для нее все уменьшается. В основном туда переносятся исправления обнаруженных ошибок, однако иногда проскакивают обновления драйверов. Марсело также выпускает `-pre`-версии и `-rc`-версии следующих релизов. Между ними есть одно важное различие: `-pre` — это ранние версии, еще не до конца протестированные, и в дальнейшем в них могут вноситься некие серьезные изменения. Версии `-rc` уже значительно стабильнее. Более того, начиная с первой версии идет чистая стабилизация ядра, то есть принимаются только исправления различных ошибок. Как правило, последний кандидат в релиз после некоторого тестирования становится финальным релизом без каких-либо изменений в коде.

2.4.x — это одна из основных веток, так что за ней тоже следует идти на <http://kernel.org>

| 2.4.x-hf |

Также стоит сказать сразу про единственное, на наш взгляд, заслуживающее внимания ответвление 2.4 — патчи `-hf` (hot-

fix) от Вилли Торрью. Хотя называть их ответвлением не совсем корректно, все патчи, появляющиеся в -hf, попадают в следующий релиз 2.4.x. В этой ветке собираются исправления серьезных ошибок и безопасности, которые с ее помощью становятся доступны раньше, чем выход следующего релиза 2.4.x. Следовательно, это однозначно полезные патчи. Нелишним будет убедиться в их наличии для того ядра, которое вы захотите поставить. Патчи находятся по адресу: <http://linux.exosec.net/kernel/2.4-hf/>.

| 2.6.x.y |

Теперь вернемся к 2.6 и обозначим еще одну довольно новую, но тоже очень важную ветку. Как я уже говорил, 2.6 в целом не так стабильна, как 2.4, и Линус решил, было, изменить схему наименования релизов, для того чтобы их больше тестировали и, соответственно, чтобы ветка была более стабильной. От изначальной идеи Линуса отговорили (в подробности вдаваться не будем, но, на наш взгляд, справедливо), но предложили создать дополнительную ветвь -stable, которую мы теперь знаем как 2.6.x.y. Линус согласился, однако спросил, где же найти таких «suckers», которые будут ее вести? Но они нашлись: Грег Кроа-Хартман и Крис Райт добровольно заполучили это почетное звание и, судя по ветке 2.6.11.y, делают свою работу очень хорошо.

В ветке -stable появляются только те патчи, которые помогают решать конкретные проблемы пользователей. В ней нет никаких обновлений, новой функциональности, и все изменения, помимо исправления какой-либо конкретной ошибки, должны удовлетворять еще одному важному требованию — простоте. Фактически это тот же «must have», что и -hf для ветки 2.4.

| 2.6.x-pre, 2.6.x-rc |

Помимо этого стоит упомянуть -pre-, а также -rc-версии ветки Линуса. Как уже упоминалось выше, Марсело Тосатти строго разграничивает -pre и -rc, однако этого нет в ветке Линуса. В последнее время версии -pre вообще стали появляться довольно редко, а кандидаты в релиз начали реально стабилизироваться только к третьей-четвертой версии. В общем, получается, что сегодня -rc стоит использовать разве что в тестовых целях, а также тогда, когда они становятся действительно надежными. Этот момент определяет сам Линус, и на правильных сайтах об этом всегда сообщается. Версии хороши, если требуется функциональность, появившаяся совсем недавно, или какое-либо исправление драйвера, к тому же, конечно, их достаточно удобно тестировать.

| 2.6.x{-pre,-rc,-}git |

Также, что касается официальной ветки, сейчас доступны еще и версии -git. Они накладываются на -pre или -rc (а если таковых еще не было, то на последнее официальное ядро) и содержат в себе снимок разрабатываемого Линусом ядра. Git — название нового инструмента управления исходниками (точнее, основы этого инструмента — хранилища) для Linux, пришедшего на смену BitKeeper. Git изначально был написан Линусом Торвальдсом, а затем в значительной степени дора-

ботан и облагорожен Петром Баудисом. Новые версии -git автоматически появляются каждый день, так что их очень удобно тестировать (особенно выяснять, какой патч вызвал проблему), однако использовать на постоянной основе не стоит, если, конечно, на это нет особых причин.

Кстати говоря, нынче очень многие ветки ведутся с помощью git, и особо заинтересованным будет полезно почитать документ Джеффа Гаржика о работе с git: <http://lkml.org/lkml/2005/5/26/11>.

Эту ветку Линуса, равно как и 2.6.x.y, а также релизы -rc и -pre, тоже можно найти на <http://kernel.org>. Долго искать не придется — ссылка на эту ветку расположена прямо на первой странице сайта.

| 2.6.x-mm |

От относительной стабильности давайте перейдем в полную нестабильность — ветку Эндрю Мортонса -mm. Она включает в себя все возможные патчи, которые только существуют в природе. Конечно, не совсем все, но, по замечанию Линуса, она содержит «wild and wacky patches» — там они «варятся», тестируются и потом плавно переходят в основную ветку Линуса. Сегодня очень многое попадает к Линусу именно через фильтр -mm. Посему ветка экспериментальная (запросто может не скомпилироваться, хотя с i386 такое случается редко). Тестировать ее удобно, но на постоянное использование должны быть веские причины.

Впрочем, одна уже поддержка файловой системы Reiser4 и технологии FUSE (подробности см. в специальной врезке) — вполне достаточные причины.

Например, автор этой статьи использует эту ветку постоянно — как раз ради Reiser4 и FUSE, примеряя к своему будущему LFS (Linux from Scratch). Также в ней сегодня можно обнаружить массу других интересных вещей, но в основном они касаются удобств и новшеств для самих разработчиков (а как хорошо работать с inotify!). В целом эту ветку можно порекомендовать отчаянным гикам, не боящимся проблем, но желающим держаться действительно на острие прогресса, а также что-либо потестировать.

Патчи этой ветки доступны в двух видах: цельный «мегапатч» — один файл, где все лежит вместе; или россыпь патчей, входящих в -mm, в одном архиве. Последний удобен для индивидуального тестирования и/или применения. То есть, если вам нужна поддержка Reiser4, но не хочется экспериментировать с остальными -mm-патчами — пожалуйста, выкачивайте их либо отдельно (доступно для каждого релиза в отдельном каталоге) и применяйте, либо берите весь архив и доставайте необходимое. Скачать патчи можно по адресу <http://kernel.org/patchtypes/mm.html>.

| 2.6.x-mm-jedi |

Кстати говоря, к ветке Эндрю Мортонса существует еще один набор патчей -jedi. Его ведет Френк Дэнис, и, как ни странно, для принципиально нестабильного -mm это нечто вроде -stable для ветки Линуса: здесь собираются исправления различных ошибок, которые, как это часто бывает, появляются в самый момент релиза. Обычно имеет смысл посмотреть, есть ли пат-

чи -jedi для последнего релиза -mm, если вы хотите его опробовать. Заодно получите устрашающее название версии ядра. Например, не так давно пришлось работать с ядром 2.6.12-rc2-mm2-jedi1.

Архив патчей находится на сервере <ftp://ftp.c9x.org/pub/linux-kernel>.

| Подсистемы 2.6.x |

Идем далее и переходим уже к значительно менее популярным веткам. В первую очередь надо отметить многочисленные ветки разработчиков каких-либо подсистем Linux. Это ACPI, IEEE1394, I2C, USB, IA64 и многие другие. Соответствующие патчи, очевидно, будут касаться конкретных подсистем и иногда будут необходимы. Не раз приходилось видеть сообщения о том, что какое-нибудь железо работает только при наличии последнего патча, например IEEE1394, но эти патчи довольно скоро перейдут в ветку Линуса, к тому же постоянно подбираются Эндрю Мортонем в свою ветку.

Основной «точкой сбора» подсистем на сегодня, наверное, можно назвать <http://kernel.org/git/>, однако некоторые подсистемы ведутся отдельно, например SCSI: www.parisc-linux.org/cgi-bin/gitweb.pl.

| 2.6.x-ck |

Переходим к менее «официальным» веткам. Тут сразу же стоит упомянуть австралийского доктора-анестезиолога Кона Коливаса и его ветку -ck. В нее включены патчи к диспетчерам процессора и ввода/вывода, кроме того, обещается улучшенная отзывчивость системы. Ветка сама по себе стабильная (и, кстати, включает в себя патчи 2.6.x.у целиком), несмотря на то, что сам Кон рекомендует употребить немного коньяка перед загрузкой такого ядра. Это ядро я также постоянно использую для другой своей машины (старенький Celeron 488 + 128 Мбайт памяти, далее фигурирует как assam), планирую перевести на него еще одну машину, как только доберусь) и чисто субъективно с этим ядром при параллельной компиляции меньше дергается музыка. Патчи можно найти по адресу <http://ck.kolivas.org/patches/2.6>

| 2.6.x-ac |

Стоит отметить ветку Алана Кокса — -ac. В нее входят исправления ошибок плюс дополнения, в основном, касающиеся драйверов. Особенно заметны в ней, пожалуй, патчи подсистемы IDE — у Алана есть серьезные претензии к сегодняшнему ведущему разработчику IDE, Бартоломею Золнеркевичу. Это ядро также знакомо пользователям Fedora Core и RHEL, так как по умолчанию там используется именно оно. Веб-сайт — <http://kernel.org/patchtypes/ac.html>. Также ветку 2.6.x-ac можно найти на kernel.org, правда, иногда kernel.org запаздывает.

| 2.6.x-tiny |

Существует еще одна интересная ветка -tiny. Она нацелена на уменьшение занимаемого ядром дискового пространства и, самое главное, памяти. Также в ней содержатся некоторые функции, полезные для ограниченных в ресурсах configura-

ций, таких как встраиваемые системы, старые 386-е и наладонные компьютеры. На наш взгляд, это прекрасный хакерский патч, которому место в официальной ветке (хотя бы частично, что касается структуры).

Он открывает множество недоступных параметров конфигурации (хотя, конечно, никто не мешает править их в исходниках и без этого патча, но надо знать, где и как): поддержка AIO (асинхронный ввод/вывод), файловой системы sysfs (исключительно полезная вещь), системного вызова SYSENTER (аппаратная особенность процессоров Pentium II и старше, значительно ускоряющая вызов функций ядра), ptrace (контроль выполнения потока), утилиты dnotify (извещение пользователей об изменениях в файловой системе), vm86 (читайте — эмуляция DOS). А ведь если выключить поддержку режима vm86 для процессоров IA32, то можно сэкономить целых 6 кбайт памяти!!! Еще в нем же представлено несколько инструментов, интересных разработчикам ядра. Мэтт Мэколл, который, собственно, и ведет эту ветку, утверждает, что минимальное ядро -tiny способно запуститься даже на машине с 2 Мбайт памяти.

Многие из открывающихся параметров экспериментальны, так что играть с ними надо аккуратнее, но у меня это ядро работало на машине assam, и делало это вполне неплохо. Чего-либо сверхъестественного не замечено, но, полагаю, для слабых компьютеров оно может быть полезно (кстати,

Перспективные технологии

Новые веяния — уже сегодня

Reiser4 (<http://namesys.com>) не просто с легкостью выигрывает множество тестов производительности, это еще и очень интересная файловая система с точки зрения архитектуры. Можно даже сказать, что это не файловая система, а система хранения объектов — разница огромная. На деле Reiser4 лишь организует взаимодействие плагинов, которые и представляют каким-либо образом данные, преобразовывают их в те или иные объекты. Даже сам файл, как структура и каталог — всего лишь плагины в Reiser4. В подробности вдаваться не буду, так как все это заслуживает отдельной статьи. Скажем еще и про FUSE (<http://fuse.sourceforge.net/>). Эта технология реализует поддержку файловых систем в пользовательском пространстве и позволяет пользователю без проблем монтиро-

вать. Звучит, возможно, сложно, но на самом деле все довольно просто — файловая система, какая бы она ни была, в FUSE работает в пользовательском пространстве, то есть для ее работы не нужно накладывать патчи на ядро, монтировать от root или что-либо подобное. Именно поэтому с помощью FUSE сейчас реализуются самые разные и самые смелые проекты, касающиеся ФС, например SSHFS — файловая система SSH. Не поняли, как это? А знаете про sftp? Вот это — то же самое, только все файлы вы будете видеть в своем привычном пространстве имен, где вам удобно. Точно так же можно работать с архивами .tar.gz и .tar.bz2 или с FTP-соединениями, а также шифровать файлы с помощью EncFS. Монтируем куда удобно и получаем доступ. Удобно? Конечно.

такое ядро по умолчанию собирается с опцией `-Os GCC`). Данная ветка представлена в Интернете на странице <http://selenic.com/tiny>.

| 2.6.x-kj |

Упомянем также ветку `-kj` (kernel janitors — дворники ядра), она интересна разве что как средство смягчения приступов альтруизма и хороша для начинающих хакеров ядра. Собственно, она и создается начинающими хакерами под руководством более опытных товарищей. В ней происходит вычищение ядра от всевозможных остатков старых API, проводятся некие мелкие оптимизации, исправляются очевидные ошибки и прочее в том же духе. Как уже говорилось, она создается начинающими хакерами — понятно, что они лишь изучают различные подсистемы/API ядра и по этому поводу очень часто просматривают исходники, а попутно делают и что-то полезное. Приветствуется всяческое тестирование этой ветки, но обычно внутреннее тестирование в ней хорошее, да и патчи не очень серьезные, поэтому изменения из нее довольно быстро попадают в ветку `-mm`, а затем и к Линусу — в основное ядро.

Подробную информацию об этой ветке можно найти на сайте <http://janitor.kernelnewbies.org>.

| 2.6.x-RT |

Еще одним из интереснейших на сегодняшний день патчей является расширение реального времени для Linux от Инго Молнара. Этот патч содержит изменения, позволяющие использовать Linux в системах мягкого и жесткого реального времени (конечно, с некоторыми допущениями, все же основа Linux и все драйверы совсем не реального времени, и с этим приходится считаться), что в первую очередь ценно для различных встроенных применений, но также уже используется и на других системах. Например, с этими патчами работают те, кто профессионально связан со звуком (пользователям Jack настоятельно рекомендуется), потому что латентность, а точнее непредсказуемость, в обычном ядре Linux сильно мешает нормальной работе.

На сегодняшний день часть патчей из этой ветки уже интегрирована в официальное ядро — именно оттуда растет вытесняемое ядро (опция сборки ядра — `CONFIG_PREEMPT`) в текущих версиях ядра, а также огромное количество исправлений для многопроцессорных машин: введение вытесняемого ядра позволило обнаружить массу неявных гонок и блокировок в ядре. Для конечного пользователя в целом этот патч может быть интересен, поскольку снижение латентности и стабильность интересна всем, даже несмотря на то, что большая часть этих изменений не будет видна глазу. Я также пробовал это ядро на машине `assam` и не могу сказать, что обнаружил какую-то большую разницу в скорости реакции. Правда, с аудио- и видеоконтентом мне работать не приходится, и оценить все его прелести довольно трудно. Поэтому пришлось вернуть на место `-sk` с включенным `CONFIG_PREEMPT`.

Сейчас ведутся дискуссии о том, стоит ли включать этот патч в официальное ядро, и, вполне возможно, он войдет через некоторое время в ветвь Эндрю Мортонна.

Патчи можно скачать на сайте <http://redhat.com/~mingo/realtime-preempt>.

| 2.6.x-vm |

Название `-vm` не совсем официальное, однако именно оно было предложено для ветки, ведомой Риком ван Риелом. Появилась эта ветвь с подачи Ника Пиггина, который заметил, что используемый ныне в Linux алгоритм замещения страниц виртуальной памяти — весьма популярный LRU, уже порядком устарел и на современных конфигурациях работает далеко не оптимально. Соответственно, Рик взялся и уже реализовал замену LRU — алгоритм CART. На сегодня в этой ветке еще не было официальных релизов, но предварительные версии доступны по адресу www.surriel.com/patches/nonresident. Wiki-страничка ветки: <http://wiki.linux-mm.org/wiki/AdvancedPageReplacement>. Мне тем временем остается только заметить, что по еще не выясненной причине я этот патч не опробовал — серьезное упущение, буду наверстывать. Патч, конечно, экспериментальный, так что будьте осторожны.

Полагаю, впрочем, что после стабилизации этот патч довольно быстро войдет в `-mm` и далее в ветку Линуса.

| Stay tuned |

На самом деле веток ядра Linux существует гораздо больше, а уж количество патчей к ядру, витающих по просторам Сети, вовсе не поддается исчислению, но это уже совсем частные случаи. Оставаться в курсе выхода новых версий разных веток Linux можно подписавшись на LKML (Linux Kernel Mailing List). Для этого достаточно отправить письмо с темой «`subscribe linux-kernel`» по адресу majordomo@vger.kernel.org, однако надо быть готовым к трафику порядка 300 писем в день. Можно еще заглядывать на <http://kerneltrap.org> (там английский язык и периодически пропускаются релизы, хотя есть масса различной интересной информации), либо обратиться к странице <http://osrc.info>.

Ну а если вам стало интересно, что еще происходит вокруг и внутри ядра (а это иногда просто необходимо, чтобы понять, зачем существует тот или иной патч, что означает тот или иной новый пункт в конфигурации и какое новое оборудование поддерживается или будет поддерживаться), стоит начать изучение с сайта <http://kernelnewbies.org/>. Знакомьтесь с Linux!

На прощание можно порекомендовать поэкспериментировать с ядрами — получите массу удовольствия, узнаете много нового, сможете поддерживать ядро в актуальном состоянии (а это значит, что количество дырок в ядре будет стремиться к минимуму). Особенно же порекомендую тестировать различные `-gc` и `-pre`-версии — этого сегодняшнему ядру действительно не хватает, размеры патчей все растут, и управляться с ними разработчикам все сложнее, так что вспоминайте дух братства свободного сообщества, тестируйте и не стесняйтесь писать разработчикам о проблемах. Файл MAINTAINERS в корне дерева исходников Linux предназначен именно для таких случаев. Успехов в ядерном синтезе! |

30 Май - 10 Июл 2005

Категории

Любая категория

Найти сейчас

Очисти

омпьютере

ичная

ни рождения и годо

гы

ари

де

гы

ари

де

гы

гы

ари

де

де

Неделя Linux в Обнинске

ALT Linux и Институт логики проводят вторую конференцию разработчиков свободных программ на реке Протве. Цель мероприятия — наладить личные контакты между специалистами, обсудить перспективы развития свободного программного обеспечения, инициировать новые проекты.

Когда: 25-27 июля

Где: г. Обнинск, Калужская область

Информация:

www.altlinux.ru/index.php?module=conf2

LinuxWorld Russia

Крупнейшая выставка и конференция, посвященная решениям, основанным на Linux и Open Source. Ежегодно проходит в 15 странах мира, в этом году — впервые в России. Выставка ориентирована на бизнес-аудиторию.

Когда: 7-9 сентября

Где: г. Москва, Гостиный Двор на Ильинке

Информация: www.linuxworldexpo.ru

Летняя школа Linux

Проводится в рамках «Недели Linux» компанией ALT Linux совместно с интернет-университетом Intuit.ru. Цель — изучение основ и вопросы преподавания ОС Linux. Будут рассмотрены аспекты обучения по лекциям учебного курса «Операционная система Linux», вопросы инструментария (подготовка и использование площадки) и специфика использования Linux (легальность, взаимодействие с сообществом, источники информации).

Когда: 25-27 июля

Где: г. Обнинск, Калужская область

Информация:

www.intuit.ru/shop/product.xhtml?id=2493377

Ежегодный фестиваль Linux в Калужской области

Слет любителей Linux и Open Source. В программе предусмотрены следующие мероприятия: круглые столы по различным тематикам, проживание в палатках, активный отдых, конкурсы.

Когда: с 30 июля по 1 августа

Где: г. Балабаново, Калужская область

Информация: <http://lrn.ru>

Первая международная конференция Wikimania 2005

Благоприятная возможность для встречи с сообществом Wikimedia и разработчиками Wiki, которые ставят во главу угла принципы свободы распространения и доступности знаний. Цель мероприятия — позволить людям, работающим над различными Wiki-проектами, узнать больше друг о друге и наладить контакты.

Когда: 4-8 августа

Где: г. Франкфурт-на-Майне, Германия

Информация: <http://wikimania.wikimedia.org>

Linux Land на Softtool'2005

Экспозиция нескольких разработчиков на основе Open Source.

Когда: с 27 сентября по 1 октября

Где: г. Москва, ВВЦ

Информация: www.softtool.ru

Сбт/Вск

04

05

11

12

18

19

25

26

02

03

09

10

04

05

11

12

18

19

25

26

02

03

09

10

04

Оконный дуэт

Существующее на сегодняшний день ПО под Linux пока не может удовлетворить все запросы пользователей, например, в части различных CAD и программ для бухгалтерии. С другой стороны, эта ОС очень сильна в плане защищенности при работе в сети. Воспользоваться этими преимуществами и при этом продолжать продуктивную работу помогут уникальные возможности системы X Window.

Идею этой статьи мне подсказал один слушатель, присутствующий на моих курсах по Linux. Как человеку, работающему с Linux продолжительное время, эта мысль вряд ли пришла бы в мне голову. Я долго думал, а зачем ЭТО надо? Но прошло немного времени, и я начал понимать, что предложение было все же достаточно интересным и надо обязательно рассмотреть его.

С чего все начиналось? После изучения темы, посвященной системе X Window и возможности запуска X-клиентов и X-серверов на разных машинах в сети, был задан вопрос: «А можно ли сделать так, чтобы все критические с точки зрения безопасности приложения работали на Linux, а отображали свой вывод на Windows-машинах?»

Первое, что мне пришло в голову: «А зачем, если вас так волнует безопасность, устанавливать Windows? Linux как рабочая станция — вполне приемлемое и безопасное решение». На что слушатель выдвинул аргумент: у него в основном работают дизайнеры, которым требуется специфичный софт, отсутствующий в Linux. Проблемы возникают в основном с программами, позволяющими осуществлять доступ в Интернет (почтовые клиенты, браузеры и т. п.), причем установка антивирусов и обновление самой системы не очень помогают. Поэтому, когда он увидел на курсах, как приложение в Linux может физически работать на одной машине, а отображать данные на другой, причем без потери качества изображения и снижения скорости работы, ему захотелось узнать, существует ли возможность отображать данные программ, работающих в Linux, на Windows-машинах. Если да, значит все программы, соприкасающиеся со «злым и ужасным» Интернетом, будут работать на Linux — в среде, где они не так сильно подвержены вирусам и прочим возможностям взлома. И даже если их поломают, это повлияет только на Linux-машину, Windows-клиенты останутся нетронутыми. Все файлы, которые клиенты будут получать из Интернета, также станут сохраняться на Linux-машине и никуда за ее пределы не выйдут, а на Windows-машине будут только окна соответствующих программ.

Об используемых технологиях

В Unix в качестве графической оболочки применяется система X Window, имеющая множество различных реализаций — коммерческих и не совсем. Если говорить о мире Linux, то на данный момент в основном используются XFree86 и X.Org. Первая система из-за проблем с лицензированием стандартно не поставляется с некоторыми современными дистрибутивами Linux, поэтому используется X.Org.

X Window построена по принципу клиент-сервер. Но сразу хочу предупредить, что принцип взаимоотношений клиента и сервера в X Window поставлен с ног на голову, и поэтому с первого раза не все понимают, как это работает. Обычно сервером или серверным приложением считается какой-либо софт, реализующий хранение информации или вычисление какой-нибудь сложной математической задачи. Клиент в обычном понимании — это программа, позволяющая управлять информацией или получать доступ к ней.

В X Window сервер содержит драйверы видеоплаты, клавиатуры, мыши и других устройств ввода и позволяет программам-клиентам рисовать на экране монитора. Таким образом, на машине пользователя работает именно X-сервер. Клиент же, в свою очередь, отображает на нем данные. X-клиент и сервер могут общаться через сетевое соединение и, следовательно, физически располагаться на разных машинах.

Система X Window подразумевает возможность использования специальных устройств — X-терминалов. На них работает только одна программа — X-сервер. Такие терминалы могут выпускаться как отдельные устройства, но в X-терминал можно превратить и любой компьютер, даже не очень новый. Программы-клиенты могут отображать данные, в том числе и на сервере, работающем на X-терминале. Это значит, что на рабочее место можно не устанавливать полноценный компьютер, а только X-терминал (концепция так называемых «тонких клиентов»). Существуют программы, эмулирующие работу X-терминала в Windows.

Теперь об окнах в X Window. Для их отрисовки используется специальное программное обеспечение — оконный менеджер (Window Manager). Как будут выглядеть окна и как они будут себя вести, зависит от него. Оконный менеджер — это тоже фактически X-клиент, то есть для отображения окон он посылает данные на X-сервер.

И еще одна немаловажная программа, которая может использоваться в X Window, — менеджер дисплеев (Display Manager). После включения X-терминал по протоколу XDMCP (X Display Manager Connect Protocol) подключается к серверу, где работает менеджер дисплеев. Последний позволяет пользователю полноценно входить в систему на Linux-машине после запуска оконного менеджера.

Хочется сразу предупредить, что протокол, используемый для связи X-сервера с X-клиентами, потребляет достаточно много ресурсов сети, кроме того, все данные, передаваемые по этому протоколу, не шифруются. Поэтому X Window рекомендуется применять только в пределах локальной сети. Передавать данные через Интернет возможно, но крайне нежелательно. Если содержимое зашифровать при помощи Ipsec или любых других протоколов, объем передаваемых данных все равно будет слишком велик. Так что, если возникнет необходимость отображать рабочие столы компьютеров, расположенных в различных частях Интернета, обратите внимание на Real VNC — эта система оптимизирована как раз под такие задачи (www.realvnc.com).

Бесплатные X-серверы для Windows

Итак, осталось найти программы, реализующие работу X-сервера в Windows. Обращение к www.google.com с запросом «X Window server Windows» выдало в ответ много информации, но самая интересная страница находится по адресу www.nixr.ru/rlfaq/rulinux.faq-6.html. Там, в разделе «6.9 Где найти X Server для Windows?», перечисляются восемь серверов. Правда, из всего списка только один оказался бесплатным — Cygwin/X. Неужели все так плохо? Все серверы коммерческие? Оказалось, что не все. После некоторых поисков был найден еще один бесплатный, написанный на Java, — WeirdX. Мне показалось, что этого мало, и, еще немного поигравшись с Google, я обнаружил freeware-сервер X-Deep/32.

Давайте рассмотрим эти три программы более подробно.

Cygwin/X (<http://x.cygwin.com>)

Широко известная компания Cygwin разрабатывает библиотеку `cygwin1.dll`, в которой эмулируется Linux API, а также портирует на Windows набор популярных GNU-утилит. Предлагаемые утилиты (`gcc`, `ld` и т. д.) распространяются по лицензии GNU GPL, однако библиотека `cygwin1.dll` имеет свою собственную лицензию (не GPL), позволяющую использовать ее в коммерческом программном обеспечении.

Cygwin/X — это порт X Window System на различные версии ОС Windows, начиная с Windows 95 и заканчивая Windows Server 2003. Cygwin/X содержит X-сервер, библиотеку X, несколько полезных скриптов и набор стандартных X-клиентов, таких как `xterm`, `xhost`, `xclock` и другие.

WeirdX (www.jcraft.com/weirdx)

Компания JCraft специализируется на написании программ с использованием технологии Java в области коммуникаций. Большая часть программного обеспечения распространяется под лицензией GNU GPL, в том числе и сервер WeirdX. Кроме WeirdX компания распространяет еще один сервер, но под коммерческой лицензией — WiredX.

WeirdX реализует все возможности стандартного X-сервера. С ним поставляется набор утилит, расширяющих возможности X-сервера: JSch организует передачу данных по защищенному каналу; JRehex — hexes-клиент и другие.

X-Deep/32 (www.pexus.com)

К сожалению, на сайте производителя не указаны особенности данного X-сервера. Заявлено лишь о полной совместимости с релизом X11R6.5.1.

Подготовка Linux-машины

Для того чтобы X-серверы могли взаимодействовать с Linux-машиной, на ней следует:

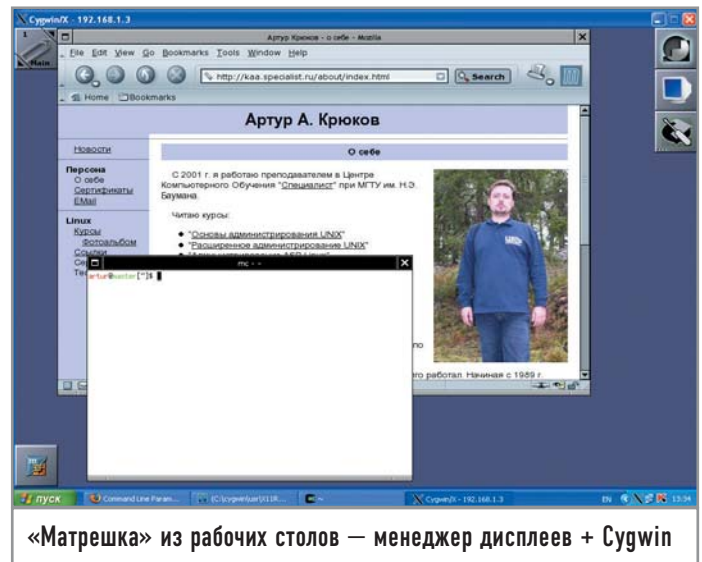
- ▶ завести учетные записи пользователей;
- ▶ если планируется использовать полноценный вход в систему (а-ля X-терминал), необходимо соответствующим образом настроить менеджер дисплеев.

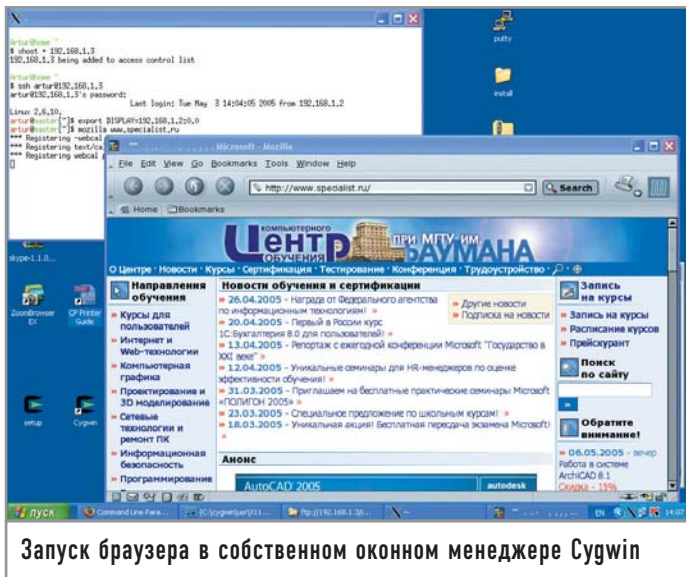
При добавлении учетной записи пользователя необходимо помнить, что у него обязательно должны быть домашняя директория и реальный shell.

С менеджерами дисплеев немного сложнее, с Linux могут поставляться три менеджера — `xdm`, `gdm` (GNOME) и `kdm` (KDE). Каждый из них настраивается различными способами. Мы будем использовать `kdm`.

Для конфигурации `kdm` необходимо:

- ▶ Найти файл `kdmrc`. В различных дистрибутивах он может находиться в разных директориях. Например, в Slackware Linux — это директория `/opt/kde/share/config/kdm`, в Red Hat Linux — директория `/etc/X11/kdm`.
- ▶ В файле `kdmrc` найти секцию `[Xdmcp]` и установить параметр `Enable=true`.





► В той же директории, где расположен `kdmrc`, находится файл `Xaccess`. В нем описываются X-терминалы, с которых возможно подключение к менеджеру дисплеев. Чтобы разрешить подключаться всем, необходимо написать символ «*» в любой пустой строке.

► В файле `/etc/X11/xdm/xdm-config`, в начале самой последней строки, содержащей параметр `DisplayManager.requestPort`, следует поставить символ «!». Это заставит менеджер дисплеев открыть для прослушивания порт `177/udp` и принимать по нему запросы от X-терминалов.

► Если на сервере с Linux не настроена система X Window, то есть не предполагается работать в графическом режиме, рекомендуется в файле `/etc/X11/xdm/Xservers` поставить символ комментария «#» в начале строки `:0 local /usr/X11R6/bin/X`. Тогда при запуске менеджера дисплеев не будет запускаться локальный X-сервер.

► Если менеджер не был запущен, сделайте это сами с помощью команды `kdm`. В противном случае заставьте его перечитать свой конфигурационный файл — `killall -HUP kdm`.

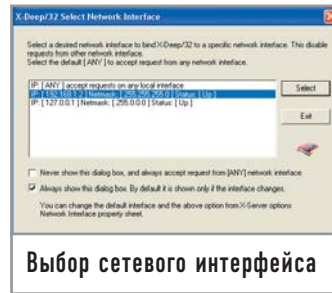
► Убедитесь, что `kdm` открыл на прослушивание 177-й порт — `netstat -nl | grep :177`.

Установка и конфигурация Cygwin/X

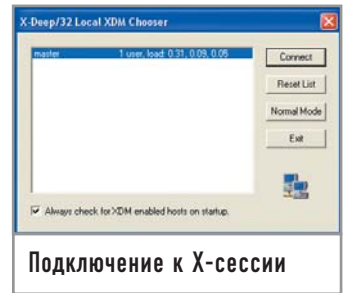
Прежде всего, скачайте с сайта разработчиков программу установки `setup.exe`. Затем можно выбрать, установить ли программу прямо из Интернета, только скачать ее или взять заранее скачанную программу из директории. Далее следует выбрать папку, куда она будет установлена и тип текстовых файлов — Unix или DOS. Затем в предложенном списке FTP-серверов необходимо найти ближайший к вам сервер. Попытка выбрать сервер в зоне .RU не увенчалась успехом, поэтому придется пользоваться услугами зарубежных серверов.

Запустите установочную программу и выберите желаемые пакеты. Также не забудьте в разделе «Net» включить установочную программу `openssh` и `openssl`.

После установки Cygwin/X необходимо настроить. Конфигурационные параметры передаются Gygwin/X при запуске



Выбор сетевого интерфейса



Подключение к X-сессии

программы `XWin.exe` в ее командной строке. Для удобства предоставляется специальный файл `startxwin.bat`.

Если при установке в качестве основной директории была выбрана `C:/cygwin`, то файл `startxwin.bat` будет находиться в директории `C:/cygwin/usr/X11R6/bin`. В нем необходимо найти строку `run XWin -multiwindow -clipboard -silent-dump-error`, в конце которой дописать `-xkbrules xorg -xkbmodel pc104 -xkblayout «us,ru» -xkbvariant winkeys -xkboptions «grp:alt_shift_toggle»`. Добавленные параметры позволяют использовать русскую клавиатуру и по своему значению полностью совпадают с аналогичными параметрами конфигурационного файла X-сервера `xorg.conf` в Linux.

Работа с Cygwin/X

После установки программы на рабочем столе создается ярлык `Cygwin`. Если его запустить, активируется командная оболочка `bash`, и вы получаете знакомую командную строку Linux. При помощи нее можно запустить X-сервер в режиме X-терминала: `X -query 192.168.1.3`. Вместо IP-адреса `192.168.1.3` следует указать IP-адрес компьютера, на котором работает менеджер дисплеев `kdm`. На экране должно появиться приглашение входа в Linux-систему.

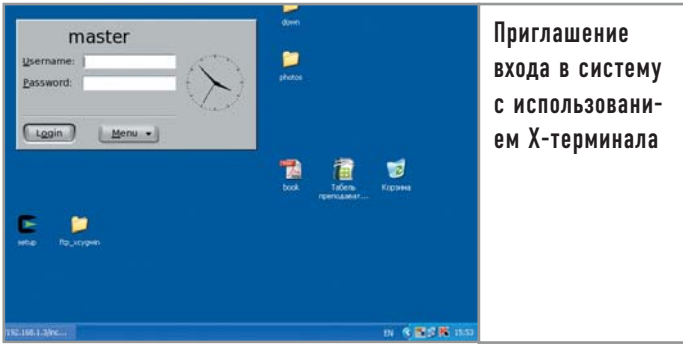
После входа в систему запускается оконный менеджер, и вы начинаете работать на Linux-машине с отображением данных в Windows.

Конечно, при использовании такого метода работы на сервере происходит большой расход ресурсов. Я лично не пробовал большое количество клиентов, но машина `dual Pentium III 800 МГц` с `1,5 Гбайт` оперативной памяти свободно обслуживала `10` рабочих мест, и это при том, что в качестве оконного менеджера выступал далеко не самый «легкий» KDE.

`Cygwin/X` позволяет использовать другой режим работы, когда на Windows-машине запускается X-сервер, собственный оконный менеджер и эмулятор терминала. В этом случае для запуска используется файл `startxwin.bat`. Благодаря параметру `-multiwindow`, который передается при запуске `XWin.exe`, каждое приложение будет запускаться в своем собственном окне.

При использовании `startxwin.bat` не происходит подключения X-сервера как X-терминала, и поэтому на Linux-машине нет необходимости настраивать менеджер дисплеев.

После запуска `startxwin.bat` появляется окно эмулятора терминала. В первую очередь в этом окне необходимо разрешить удаленным программам отображать данные на X-сервере, работающем в Windows. Для этих целей воспользуйтесь программой `xhost`: `xhost +192.168.1.3`. Вместо показанного в примере



IP-адреса следует использовать IP-адрес машины Linux, на которой будут запускаться программы.

Зайти на Linux-машину можно при помощи программы ssh: `ssh user@192.168.1.3`. Вместо указанных в примере пользователя и IP-адреса следует использовать логин пользователя на Linux-машине и, соответственно, ее IP-адрес.

Теперь необходимо сделать так, чтобы программы, запускаемые на Linux, отображали свои данные в окне Windows. Один из вариантов решения — использование переменной среды окружения `DISPLAY`: `export DISPLAY=192.168.1.2:0.0`. Вместо IP-адреса, показанного в примере, следует использовать IP-адрес компьютера, на котором работает X-сервер. Программы запускаются обычным образом.

Установка и настройка WeirdX

Сервер поставляется в виде архива ZIP, который достаточно распаковать в любую директорию. Перед запуском сервера в систему необходимо установить виртуальную машину Java как минимум версии 1.1.

В директории, куда был распакован архив, появится папка `config`, в которой будет находиться всего один конфигурационный файл — `props`. Все параметры сервера задаются именно в этом файле.

Во-первых, в файле `props` следует найти параметр `weirdx.display.acl` и определить, кому можно отображать данные на этом X-сервере. Например: `+192.168.1.3` или просто символ «+», если доступ разрешен всем.

Запуск сервера лучше производить при помощи специального пакетного файла, расположенного в директории `misc`, — `weirdx-Java2.bat`. После запуска X-сервера необходимо зайти на Linux-сервер при помощи любой программы, поддерживающей протокол ssh (например, PuTTY). Как и в случае с Cygwin/X, необходимо определить переменную `DISPLAY` и запускать программы.

При работе WeirdX очень сильно притормаживал (сказалось использование Java), и при отрисовке на экране появлялись черные полосы. В общем, качество данного продукта сильно разочаровало. Поэтому я даже не буду приводить скриншоты, демонстрирующие его работу.

Установка и настройка X-Deep/32

На сайте X-Deep/32 не было обнаружено никакой документации по программе кроме FAQ. Установочный файл занимает почти 16 Мбайт. Необходимый объем свободного дискового пространства — 43 Мбайт.

Инсталляция представляет собой обычную процедуру установки программы Windows. После установки были обнаружены документация по продукту и две программы — X-сервер и X Client Launcher.

После запуска появляется диалоговое окно, предлагающее выбрать сетевой интерфейс, на котором X-сервер будет «слушать» запросы. Затем программа сама пытается найти доступные в сети менеджеры дисплеев. После обнаружения выводится список таких серверов.

С первой попытки подключиться к менеджеру дисплеев не удалось — в сети необходимо настроить корректное преобразование имен. Но после запуска X-Deep/32 можно открыть окно настроек и там явным образом указать Linux-машину, к которой следует подключаться.

При всей простоте установки и настройки X-сервер показал себя не с лучшей стороны в плане быстродействия. Cygwin/X работает гораздо быстрее. Встроенная в X-Deep/32 возможность удаленного запуска программ базируется на `rsh` и `gexes`. Ни один здравомыслящий администратор эти программы использовать не будет. Так что в случае X-Deep/32 остается только режим эмуляции X-терминала, что не всегда удобно.

Итоги

В результате проверки трех бесплатных X-серверов под Windows, по моему мнению, наиболее приемлемым является реализация от Cygwin. Несмотря на то что X-сервер требует установки почти всей базовой среды запуска, работает он просто великолепно. И это не удивительно, ведь он представляет собой стандартную реализацию сервера X.Org, собранную в среде `cygwin`. Соответственно, для конфигурации сервера используются те же приемы, что и в Linux.

Про WeirdX ничего хорошего сказать, увы, нельзя. Во-первых, он жутко «тормозил». Во-вторых, для запуска приложений требуется дополнительная программа, при помощи которой можно подключиться к Linux-машине. В общем, медлительность Java сказывается при написании приложений пользователя.

X-Deep/32 мог бы стать чемпионом, если бы в нем была нормально реализована возможность запуска программ — или хотя бы встроенный ssh-клиент для удаленного логина на Linux-машину.

Возможно, существуют и другие бесплатные реализации X-серверов для Windows. Хочется надеяться, что платные варианты будут демонстрировать большую скорость работы и легкость конфигурирования, ведь не зря их продают за деньги (цены колеблются от \$40 до \$300 за рабочее место). Но лично я считаю, что держать рабочие места под управлением Windows, для того чтобы несколько человек могли запускать специфичные для этой ОС программы, — это неудобно. Проще выделить один терминальный сервер под управлением Windows 2000, на котором будут установлены программы типа 1С, системы CAD и прочие. А на рабочих местах лучше всего использовать бездисковые рабочие станции под управлением Linux и запускать на них терминальный клиент для Windows — `rdesktop` (www.rdesktop.org).

Выражаем благодарность Центру компьютерного обучения «Специалист» при МГТУ им. Н. Э. Баумана за помощь в подготовке материала

Системный анабиоз

Многие пользователи уже давно знают про одну из интересных особенностей стандарта ACPI (acpi.sourceforge.net) — так называемые sleep states, или «состояния сна». Они описывают различные режимы энергосбережения, в которые можно переводить ACPI-совместимый компьютер. Кроме обычных и оставшихся со времен АРМ, вроде отключения дисков или перевода устройств в режим «полусна», есть несколько новых и самых интересных: это ACPI S3 (или `suspend-to-ram`) и ACPI S4 (или `suspend-to-disk`). Как правило, еще в ходу сокращения STR для S3 и `hibernate` для S4. Мы рассмотрим эти состояния более подробно.

STR, или `suspend-to-ram`

Это состояние, при котором все компоненты компьютера переводятся в режим максимально низкого энергопотребления (либо вообще отключаются), но питание для оперативной памяти сохраняется — таким образом, не происходит ее разрядка, и содержимое остается прежним. При включении компьютер делает лишь только повторную инициализацию отключенных или «усыпленных» устройств, не производя полной загрузки системы.

S4, или `suspend-to-disk` aka `hibernate`

Самый интересный режим, позволяющий «заморозить» состояние всей системы, сохранить его на диск и полностью отключить компьютер. Этот режим, который, в отличие от STR, все-таки зависит от состояния электросети, позволяет «спать» компьютеру сколь угодно долго — до тех пор, пока диск не развалится.

Вернемся к пользователям Windows, они знают этот режим под названием `hibernate`, вариации которого можно изменять либо из BIOS (для STR), либо через средства управления самой Windows. Как правило, востребовано все это на переносных компьютерах: вроде как выключил и забыл, потом пришел домой и продолжил работу с того места, где закончил в офисе. Но и для обычных домашних пользователей это полезно — ведь на «засыпание» тратится гораздо меньше времени, чем на за-

грузку с нуля. Здорово, скажете вы, но причем тут Linux? А вот причем: в ней тоже это есть. Поддержка sleep states была добавлена в ядро 2.4 и серьезно улучшена в версии 2.6. Теперь там тоже поддерживаются спящие режимы, причем в несколько более полном варианте.

Расширенное использование

Особые возможности

Существует специальный скрипт, облегчающий подготовку компьютера к «усыпленнию». Он находится на сайте `swsusp2`, называется просто и незатейливо `hibernate` и представляет собой набор функций и полезных трюков. Например, с его помощью можно вызвать программу для восстановления видеорежима после пробуждения, настроить параметры сохранения через привычные названия опций, а не через `echo` «видеорежим» >

`/proc/software_suspend2`, выполнить какие-то действия перед засыпанием/после пробуждения и т. д. Вместе с этим скриптом можно использовать популярную программу `acpid`, которая умеет обрабатывать действия нажатия кнопок на корпусе (или датчика закрытия крышки у ноутбука). В общем, `hibernate` — очень полезная во всех отношениях программа, которая рекомендована каждому, кто будет пользоваться `swsusp/swsusp2`.

| Swsusp, или первая волна |

Для начала немного скучной теории: чтобы компьютер смог сделать S4, необходимо каким-то образом записать «замороженное» состояние на диск. У Windows и большинства ноутбуков с этим проблем нет — нужна всего лишь «правильная» BIOS, умеющая делать запись на специальный раздел диска (как правило, его нельзя увидеть из Windows) и корректно выключать всю систему «по приказу» из Windows.

Для Linux подобный механизм работает плохо — во-первых, «неправильных» BIOS все-таки больше, да и знает он про Linux немного. Поэтому максимум, на что можно рассчитывать, — это вполне корректное выключение/усыпление устройств, и не более того. А все остальные задачи придется решать программно и своими силами. Прибавьте к этому обилие компонентов ядра, изначально не спроектированных на «усыпление», драйверы устройств, в которых отсутствует поддержка восстановления после «просыпания», и т. д. В общем, картина не совсем приятная. Часть этих проблем может решить созданный механизм под названием Swap Suspend, или swsusp, который позволяет выполнять большинство операций (остановка всех выполняемых процессов ядра, запись состояния регистров процессора, перевод драйверов в энергосберегающий режим), а также записывать «замороженный» образ в раздел, предназначенный для своппинга (или подкачки — говоря языком пользователя Windows). К сожалению, первые версии swsusp работали крайне нестабильно: это было связано с плохим тестированием (просто не было достаточной аппаратной базы), да и пользователи особого интереса не испытывали, потому что пользователи могут появиться только у стабильно работающего решения.

| Swsusp2, или революционный метод запуска вашего компьютера |

К счастью, добрые люди не дали умереть swsusp, и появился форк от основного swsusp под названием Software Suspend 2, или swsusp2 (www.suspend2.net). Адрес Wiki по swsusp2 — <http://wiki.suspend2.net>. Все части оригинального «заморажи-

Варианты использования

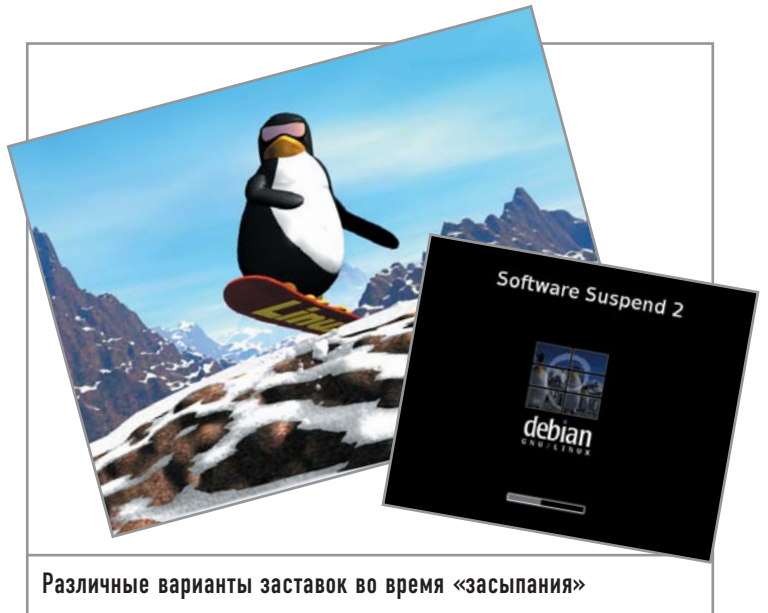
Когда дорога каждая секунда

► **Сервер приложений с критичным временем доступа и важными данными. Вдруг вам понадобилось срочно остановить сервер, и у вас нет времени на его нормальное завершение (например, осталось 30 секунд до окончательной разрядки UPS). Без проблем — «усыпляете» сервер, выключаете его, потом снова включаете, когда неприятности позади, и сервер снова работает!**

► **У вас старый ноутбук, у которого никогда не будет**

STR или S4. Ставите swsusp2 и «усыпляете» его, когда хотите, так как swsusp2 больше зависит от ядра, а не от версии BIOS.

► **Нужно срочно заменить вышедший из строя компонент (например, сетевую плату), а у вас сервер с критичным временем доступа (предположим, там расположен хостинг или почта). «Усыпляете» его, заменяете сетевую плату, и, если вы все сделали быстро, никто ничего и не заметит.**



Различные варианты заставок во время «засыпания»

вателя» подверглись серьезной доработке: было затронуто само ядро (в частности, планировщик процессов и различные ключевые компоненты), код многочисленных драйверов (видеоадаптеры, сетевые платы, устройства мультимедиа и т. д.). Появилась возможность сжимать образ памяти перед сохранением, подгонять его под размер свободного места на диске (в документации к swsusp2 этот процесс называется поеданием памяти), стало возможным использовать различные плагины для режимов сохранения (swapwriter, filewriter), появилась поддержка boot splash (крутящийся на экране во время «засыпания» пингвин на сноуборде — это гораздо интереснее, чем синий экран Windows). И самое главное — можно «усыплять» не только ноутбуки, но и серьезные серверы, так как теперь поддерживаются SMP и HIGHMEM (то есть количество памяти превышает 1 Гбайт)! Есть патчи для поддержки PPC и x86-64, но пока они не признаны очень стабильными. На сегодняшний день осталось еще много недостатков в работе swsusp2, но большинство из них можно решить — например, не пользоваться закрытыми коммерческими драйверами (для видеоплат ATI или NVIDIA) или выбирать компоненты от более-менее известных производителей, так как гораздо больше шансов, что, например, «кривая» BIOS быстрее исправит ASUS, чем Elitegroup.

| Заключение |

Если вы заинтригованы и решили проверить, как работает swsusp2 на вашем компьютере с дистрибутивом Linux, к тому же этот дистрибутив — от ALT Linux (например, Master-2.4), то вам крупно повезло. Так как для него ничего дополнительно собирать или скачивать не надо, достаточно поставить из backports специальное ядро (из серии -wks26) и сам скрипт управления hibernate. Если в вашем дистрибутиве таких возможностей нет, обратитесь к документации на сайте swsusp2 или расположенному там же списку рассылки для пользователей. А если вы разработчик и желаете помочь проекту своими патчами или отчетами о замеченных ошибках, присоединяйтесь к обществу создателей swsusp2. В любом случае вы поможете и себе, и другим. |

Раздолье в серверном заповеднике

Современные компьютеры становятся все быстрее и быстрее, и грех этим не воспользоваться. Однако и тут есть свои подводные камни. Предположим, у вас мощный компьютер с большим объемом оперативной и дисковой памяти, и вы хотите создать мощный сервер, предоставляющий множество различных сервисов. «Без проблем», — скажете вы — и будете не правы, потому что нужно еще сделать так, чтобы сервер можно было легко администрировать и чтобы само решение было максимально безопасным.

| Что это и кто это? |

На сегодняшний день существует много технологий, позволяющих сделать жизнь администраторов спокойнее. Это и изоляция сервисов (chroot), и защита от переполнений буфера (owl или grsecurity), и разграничение привилегий (например, LIDS или RSBAC). Но согласитесь, было бы все же лучше, если бы некоторые небезопасные сервисы можно было физически перенести на другой сервер (например, сервисы баз данных, различные IM-сервисы и т. д.). «Как, — скажете вы, — у нас уже есть один сервер, самый мощный и лучший, зачем нам еще?» Хорошо бы сделать так, чтобы на базе одного физического сервера можно было создать несколько виртуальных и на каждый из них установить свой дистрибутив и настроить его по вкусу.

Видимо, подобные мысли не давали покоя программисту и администратору Жаку Глина, и он придумал решение, которое

было названо VServer (изначально оно носило имя Virtual Private Server, но когда так стали называть другой проект, было принято решение о переименовании). По сути, это сервер внутри сервера — со своими ресурсами, сетевыми интерфейсами, памятью, ограничениями и пользователями. При этом серверы не видят и не влияют друг на друга. Свежую версию VServer можно найти в Интернете по адресу www.linux-vserver.org.

Кому, кроме администраторов, это решение может пригодиться? Говоря словами авторов, «всем, кому нужно управлять сервером». Вот несколько примеров, показывающих, в каких областях это применимо:

- ▶ Хостинг. Полностью независимый от ограничений провайдера хостинг (то есть несколько отдельных виртуальных серверов в пределах одного физического).
- ▶ Эксперименты с программным обеспечением. Согласитесь,

не совсем правильно производить отладку или разработку программного обеспечения на рабочей системе, а вот делать это на отдельной машине было бы вполне логично.

- ▶ Образовательные цели. Например, каждому студенту можно дать свой сервер, с которым он сможет делать все, что захочет. Можно даже дать ему пароль root.
- ▶ Изолированные серверы. Например, если нужно работать с каким-то приложением и необходим полный контроль за его действиями.
- ▶ «Сервер-приманка» (honeypot). Незаменимая вещь для защиты и исследования сети.
- ▶ Поддержка работы нескольких версий серверов и возможность их независимого включения и отключения.

Кстати, и кошмар, связанный с резервным копированием, может уйти в прошлое, ведь теперь сервер — это всего лишь каталог на диске, который можно скопировать когда и куда угодно. А основная система — всего лишь минимальный набор пакетов и ядро с vs-патчем. В системах промышленного уровня раздел с виртуальными серверами может храниться на LVM, который находится поверх DRBD и т. д. Если вы используете файловую систему xfs, можно делать «снимки» раздела и сохранять их на устройстве для резервной записи — в общем, есть где разгуляться фантазии настоящего администратора. В случае хостинга есть тоже одно неоспоримое преимущество — реальный биллинг, который можно гибко настраивать в пределах одной системы (ведь у каждого сервера только один IP, и его невозможно поменять или подделать).

Архитектура VServer

Linux VServer состоит из двух частей:

- ▶ Патч к стандартному ядру Linux. Он добавляет поддержку контекстов безопасности, воплощающих основную идею VServer. Они позволяют изолировать каждый VServer внутри основной системы и исключить их взаимное влияние друг на друга. Процесс, выполняющийся в рамках определенного контекста, может видеть лишь такие же процессы и использовать только IP-адрес, присвоенный данному контексту. Процессам запрещено создавать устройства и открывать прямой доступ к ним (при этом данные ограничения действуют даже для суперпользователя).
- ▶ VServers Tools. Набор программ для создания и управления виртуальными серверами. В основном все они написаны на языке командной оболочки, что позволяет их легко модифицировать под свои нужды. В том числе в VServers Tools включена и программа VServer, позволяющая запускать, останавливать и заходить в виртуальный сервер.

Изолированные области

Все системы на базе Linux или Unix имеют специальный вызов — chroot(). Он требуется в том случае, если нужно «посадить» процесс в отдельную директорию. После этого процесс, для которого был сделан chroot(), будет считать, что директорию, в которой он находится, является корневым каталогом. Вызов chroot() нельзя отменить. Единственное, что остается делать процессу в этой ситуации, — это отправляться в дирек-

тории уровнем ниже, повторяя chroot() снова и снова. Вы, наверное, уже догадались, к чему шло все это вступление: основная идея технологии VServer — системные вызовы, позволяющие полностью изолировать запуск и исполнение процессов в определенных местах внутри сервера.

Виртуальный сервер изолирован в четырех областях:

- ▶ Файловая система. Каждый VServer «посажен» в отдельную директорию главного сервера и не может «убежать» из нее. Это ограничение реализовано через вызов chroot(), имеющийся в любой Unix-совместимой системе.
- ▶ Процессы. VServer может видеть только те процессы, которые находятся в одинаковом с ним контексте безопасности. Даже сервер с контекстом root может видеть только свои процессы (это сделано для того, чтобы он был более безопасным в использовании). Для просмотра всех процессов на всех виртуальных серверах существует специальный механизм (контекст с номером 1).
- ▶ Сеть. Каждый VServer имеет уникальные имя и IP-адрес, которые могут быть использованы сервером для сетевых соединений и ответов на запросы. Кроме того, это ограничение заметно для VServer.
- ▶ Суперпользователи и их возможности. Суперпользователь, работающий с VServer, имеет меньше привилегий по сравнению с тем, который работает в обычной системе Linux. Например, он не может настраивать и отключать сетевые интерфейсы и многие параметры системы, а также подсоединять или отсоединять файловые системы, работать с блочными устройствами и т. д. Грубо говоря, суперпользователь VServer имеет полный доступ только к тому, что в нем есть, а именно — к файлам и процессам, и большего ему не дано.

Взаимодействие между процессами (System V IPC)

Все внутренние IPC-ресурсы изолированы от IPC-ресурсов других VServer. Для выбора необходимого в качестве ключа используется номер контекста безопасности каждого VServer.

```

root@beehive2 root# vserver-stats
  CTX  PROC  VSZ  RSS  userTIME  sysTIME  UPTIME  NAME  DESCRIPTION
  0     42  49KB  5KB  9w25.09  2w41.05  6h17m24  root server
  101    3    4KB  617B  m88.88  m88.88  m27.91  trustix15
  102   52  726B  2KB  7w52.25  2m89.68  5h7m38  master.localoffice
  103   63  511KB  93KB  12w32.55  m39.94  5h30m43  cedar.elkotel.ru
  104   14   36KB  3KB  m88.47  m88.22  4h19m47  master24
  105    3    6KB  609B  m88.88  m88.88  m48.18  trustix21
  
```

Работа утилиты vserver-stats: статистика активных виртуальных серверов; на заднем плане ssh-сессия с одним из VServer

```

lakostis@lks: ~/home/lakostis
790 0 MAIN tty6 00:00:00 mingetty
791 0 MAIN ttyS0 00:00:00 mingetty
794 0 MAIN ? 00:00:00 distccd
795 0 MAIN ? 00:00:00 distccd
796 0 MAIN ? 00:00:00 distccd
017 0 MAIN ? 00:00:00 sshd
022 0 MAIN ? 00:00:00 sshd
023 0 MAIN pts/0 00:00:00 bash
047 0 MAIN pts/0 00:00:00 su
051 0 MAIN pts/0 00:00:00 bash
1445 0 MAIN ? 00:00:28 kjournald
1580 182 monster.localoffice ? 00:00:10 named
1593 182 monster.localoffice ? 00:00:00 crond
1629 182 monster.localoffice ? 00:00:03 xinetd
1653 182 monster.localoffice ? 00:00:00 sshd
1678 182 monster.localoffice ? 00:00:00 libhttpd.ep
1693 182 monster.localoffice ? 00:00:00 libhttpd.ep
1694 182 monster.localoffice ? 00:00:00 libhttpd.ep
1695 182 monster.localoffice ? 00:00:00 libhttpd.ep
1696 182 monster.localoffice ? 00:00:00 libhttpd.ep
1697 182 monster.localoffice ? 00:00:00 libhttpd.ep
1746 182 monster.localoffice pts/0 00:00:00 mysqld_wrapper
1814 182 monster.localoffice pts/0 00:04:45 mysqld
lines 29-51

```

Список процессов на «родительской» системе вместе с запущенными процессами на VServer

Это позволяет создать удобную и гибкую среду для работы виртуального сервера. И каждая из перечисленных возможностей может быть использована независимо от других.

Безопасность

Даже если вами было создано окружение, в котором процессы ограничены лишь собственным виртуальным миром и общаются с сетью только через выделенный IP-адрес, вы должны сделать так, чтобы и ущерб, который могут вам причинить эти процессы, был минимальным. Для этого нужно создать виртуальные среды и дать им ограниченные привилегии root. Как же можно ограничить процессы с правами root, которые обладают таким контролем над системой (например, могут просто перезагрузить компьютер)? Введите систему возможностей (capabilities). Идея не нова, но мы считаем, что очень многие пользователи никогда не слышали о существовании подобной системы.

В старые времена систем Unix/Linux пользователь root (обладающий идентификатором 0) имел возможность делать такие вещи, о которых другие пользователи даже не могли и мечтать. Весь доступ к ядру, системным вызовам или просто некоторым ресурсам прекращался, если идентификатор пользователя (или процесса, запущенного этим пользователем) не был равен нулю. Для того чтобы процесс с ID 0 мог хоть немного понизить привилегии, существовал только один путь — менять идентификатор на отличный от нуля. Одним словом, либо все, либо ничего. И вот тут-то и появилась система возможностей.

Система возможностей

Сегодня все, что отличает root от других пользователей, — это набор возможностей. Пользователь root обладает всеми возможностями, а другие — нет; ID 0 больше уже ничего не значит — просто красивая цифра. На данный момент насчитывается около 30 возможностей. Можно сделать так, чтобы процесс утратил какие-либо возможности навсегда и они к нему больше никогда не вернулись.

Возможности позволяют уменьшить силу root — и это именно то, что нужно, если вы хотите создать собственного

суперпользователя. Процесс с его правами, запущенный внутри виртуального окружения, к примеру, может открыть порт перед 1024, но окажется не в состоянии отключить или перенастроить параметры сетевого интерфейса. Рассмотрите внимательно файл `/usr/include/linux/capability.h` — и вы увидите, что это вполне реально.

Для VServer было придумано несколько новых возможностей: например, `new_s_context`, позволяющая создать новый контекст, а также `set_ipv4root`, способствующая «привязке» всех процессов внутри контекста к одному IP-адресу. Кстати, все эти ограничения распространяются и на потомков процессов. Однако новые возможности нельзя отключить обычным способом, то есть мы «посадили» VServer в `chroot/s_context/ipv4root`, но «убеждать» его процессы оттуда уже не смогут. Таким образом, используя существующие и новые системы возможностей, мы можем сделать для VServer некую среду, где root будет ограничен в своих правах и не сможет влиять на работу основного сервера.

Для работы с системой возможностей в состав `vserver-utils` входят утилиты `reducescap` и `chcontext`. Мы продемонстрируем, как функционируют они и работает сама система возможностей, на конкретном примере.

Работаем с контекстами

Утилита `chcontext` позволяет зайти в определенный контекст безопасности, выделенный для VServer, и запустить программу, указанную в качестве аргумента, с правами, ограниченными этим контекстом. Таким образом, эта программа уже не сможет видеть процессы, запущенные на основном сервере.

Например, запустим два окна с приложением (пусть это будет `xterm`), с одинаковыми ID пользователя.

В первом окне зададим команду:

```
xterm
```

Во втором окне выполним `chcontext`, запускающую командную оболочку. Затем попробуем ввести `ps` и обнаружим, что видно нам совсем немного. Потом попробуем «убить» `xterm`, находящуюся в первом окне, и убедимся, что этого сделать нельзя. Теперь выходим из оболочки и возвращаемся к привычным возможностям.

Во втором окне укажем следующее:

```
/usr/sbin/chcontext /bin/sh
```

```
ps
```

```
killall xterm
```

```
exit
```

Таким же образом можно порождать внутри контекстов другие контексты, которые будут к тому же изолированы друг от друга.

Понижаем возможности

Утилита `reducescap` позволяет ограничить количество возможностей процесса и его subprocessов. Даже программа `setuid` не сможет увеличить число возможностей, ограниченное с помощью `reducescap`.

Рассмотрим это ниже на конкретном примере:

```
# Мы сейчас не с правами root
```

```
# Посмотрим текущий потолок возможностей
```

```

cat /proc/self/status
# Строка capBset показывает в основном 1
/usr/sbin/reducecap --secure /bin/sh
cat /proc/self/status
# capBset теперь показывает гораздо больше 0
# capEff теперь вся из 0, то есть у нас нет больше никаких
привилегий
# теперь попробуем стать root
su
cat /proc/self/status
# saEff содержит теперь гораздо меньше 0
# но попробуем проверить, root ли мы на самом деле
tail /var/log/messages
# это мы можем
/sbin/ifconfig eth0
/sbin/ifconfig eth0 down
# а это уже нет, таким образом, мы потеряли свои возможности
# супермена
exit

```

Работаем с VServer

Итак, после небольшого вступления попробуем поработать с VServer. Для этого понадобится дистрибутив Linux (на базе RPM или APT), ядро с vs-патчем и набор vsserver-utils.

В дистрибутиве ALT Linux это может выглядеть таким образом:

```

# ставим ядро с поддержкой VServer
$ apt-get install kernel-image-vs-smp
# ставим утилиты для работы с VServer
$ apt-get install util-vsserver

```

Теперь создадим наш маленький VServer:

```
$ vsserver build <name>
```

В данном случае <name> — это произвольное имя виртуального сервера.

Далее утилита VServer создает структуру будущего виртуального сервера на базе текущего дистрибутива основного сервера.

Теперь зайдём в наш новый виртуальный сервер:

```
$ vsserver enter <name>
```

Здесь <name> — также произвольное имя виртуального сервера.

После выполнения этой операции получаем стандартное приглашение командной оболочки, как будто бы мы зашли на этот сервер с клавиатуры. Редактируем конфигурацию по вкусу, не упуская из вида тот факт, что теперь у нас один IP-адрес — и все службы, работающие на этом сервере, должны быть привязаны только к нему.

Теперь запускаем наш VServer:

```
$ vsserver start <name>
```

И снова <name> — произвольное имя виртуального сервера.

Теперь этот сервер доступен другим и может отвечать на запросы извне (например, если запущен ssh, мы можем управлять им по ssh). Если возникнет необходимость остановить этот сервер, нужно просто набрать следующее:

```
vsserver stop <name>
```

В процессе создания VServer одноименная утилита записывает в каталог /etc/vservers его конфигурацию, которая состоит из двух файлов: первый описывает общие параметры

(номер контекста, IP-адрес, включенные возможности), а второй — порядок действий, выполняемых на этапе запуска/остановки. Для удобства обслуживания нескольких виртуальных серверов существует скрипт init.d, позволяющий производить их запуск или остановку выборочно. Существуют еще и утилиты для унификации содержимого VServer, просмотра процессов, остановки/перезагрузки, а также многие другие. Все они входят в состав util-vsserver или могут быть написаны самостоятельно, поскольку в основном это скрипты на языке командной оболочки.

Совместимость с другими security-патчами

Предположим, вы создали виртуальный сервер и убедились, что он работает, но все же чего-то не хватает. Все правильно, полной уверенности не бывает. Ведь вы оградил себя системой привилегий, возможностей, заперли всех в контексты, но забыли самое главное — защиту «родительской» системы от самой себя и собственных ошибок. Переполнение буфера, ненадежные права доступа, низкое качество кода выполняемых программ и ошибки ядра — все это осталось по-прежнему с вами: общие меры по защите никто не отменял. Обычно применяют патчи к ядру, исправляющие его код и добавляющие улучшения, связанные с безопасностью (например, патч Openwall для ядер 2.4 и патч PaX для ядер 2.6). Также необходимо соблюдать общие принципы безопасного администрирования — запрещать то, что не разрешено, и не включать то, что не нужно.

Аналоги и аналогии

Если вы хотите объяснить собеседнику что-то, о чем он не имеет совсем никакого представления, вы наверняка прибегаете к аналогиям. При описании VServer тоже сложно удержаться от этого метода повествования. Кажется, все, о чем здесь идет речь, вы уже где-то видели или слышали. Да, общие идеи витают в воздухе, поэтому неудивительно, что они приходят в головы разных людей почти одновременно. Мы

Накладываем ограничения

Квоты и лимиты

На сегодняшний день для VServer есть несколько патчей (некоторые из них уже вошли в нестабильную версию VServer для ядра 2.6), позволяющих лимитировать ресурсы, которые потребляются, во-первых, самим сервером, а во-вторых, пользователем внутри него. Например, столь популярные дисковые квоты можно использовать и внутри VServer, а также задавать их

для каждого сервера в отдельности (при этом внутри VServer дисковое пространство будет ограничено данной квотой). Можно ограничивать количество запущенных процессов внутри VServer, задавать глобальный nice-уровень для всех процессов и т. д. Кроме того, есть возможность управлять доступом к каталогу /proc, что делает работу основного сервера более безопасной.

же попытаемся понять, на что похож VServer, попутно рассказывая об аналогичных технологиях.

Виртуальные машины

В чем-то VServer является виртуальной машиной, но у него нет типичных недостатков, присущих любым виртуальным машинам или эмуляторам:

- ▶ **Скорость работы.** VServer действительно работает с той же скоростью, что и обычный железный сервер, так как в данном случае ничто не эмулируется и не «подделывается». То есть, он требует не больше ресурсов, чем, скажем, VMWare или qemu. С другой стороны, в VServer не существует такого понятия как блочное устройство, но имеются раздел, память и процессор с сетевым интерфейсом: процессам обычно большего и не требуется.
- ▶ **Архитектура.** Это слабое место VServer. Нет возможности установить Windows или FreeBSD — вы ограничены архитектурой, поддерживаемой «родительской» системой. Впрочем, не стоит забывать, что это не эмулятор.
- ▶ **Ядро.** В VServer существует единое ядро для всех. Это обстоятельство можно рассматривать и как удобство (согласитесь, лучше менять ядро одновременно на всех серверах сразу, чем переставлять на каждом отдельно), и как недостаток (по сравнению, скажем, с UML или Xen, о которых речь пойдет немного ниже).
- ▶ **Управление памятью и ресурсами.** В Xen можно назначить на каждую машину даже свой планировщик, а в VServer один имеется лишь один планировщик. В тестируемой ветке для ядра версии 2.6 встречаются идеи того, как это можно сделать и для VServer.
- ▶ **Расширенный мониторинг процессов и потребляемых ресурсов.** Находится в стадии активной разработки, и скоро эти функции можно будет использовать в полной мере.

Проекты, похожие на VServer

VE — Virtual Environment

Еще в 2001 году существовал интересный проект Virtual Environment, впоследствии переименованный в Virtuozo. Разрабатывала его компания SWSOft (та самая, которая подарила нам ASPLinux). Потом проект как-то тихо перестал развиваться — вполне возможно, его вытеснил User Mode Linux. Архив VE или Virtuozo (aspcomplete/) можно найти в Интернете на странице Пола Слейдена, посвященной VServer и размещенной по адресу www.paul.sladen.org/vserver.

UML — User Mode Linux

Интересный проект, интересная идея. Можно запускать собственные ядро и систему в рамках основной системы, перегружать и отлаживать это самое ядро и многое другое. Еще до появления непосредственно VServer данный проект часто использовался в образовательных целях, а также находил применение в сфере хостинга и отладки программного обеспечения. Фантазия ограничена только Linux. Подробную информацию по данному вопросу можно найти в Сети по следующему адресу: <http://user-mode-linux.sourceforge.net>.

VPS — Virtual Private Servers

Коммерческий проект, представляющий собой ответвление VServer, целиком и полностью предназначенный для организации хостинга на базе VPS. Ресурсы, ориентированные на хостинг с помощью VServer: OpenVPS (www.openvps.org) и FreeVPS (www.freevps.com).

Xen — The Xen virtual machine monitor

По сути, это микроядро для Linux, позволяющее запускать ядра других ОС. Поддерживаются изоляция, ограничения и управление ресурсами. Довольно интересный и многообещающий проект (вполне годится в качестве замены Win4Lin или VMWare ESX/GSX, так как имеет опыт успешной эксплуатации Windows в среде Xen). Подробная информация размещена на сайте www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html.

RAD GNU/Linux

Проект, поддерживаемый Петром Савельевым, целью которого является создание дистрибутива для маршрутизаторов — простого в настройке и чем-то похожего на системы Cisco. RAD GNU/Linux является хорошим примером применения ALT Linux Sisyphus для разработки дистрибутивов. В системе используется технология VServer. Сайт: www.radlinux.org.

Заключение

В завершение хотелось бы привести цитату разработчиков VServer, чтобы лучше понять, как они представляют себе будущее этой технологии.

«Идея виртуального сервера интересна, поскольку позволяет достичь более высокого уровня безопасности и сократить количество административных задач, необходимых для его обслуживания — например, обычных операций, таких как резервное копирование и обмен данными. Службы вроде мониторинга можно настроить лишь единожды (так как физический сервер всего один).

Сервер на базе Linux может одновременно выполнять множество задач с высоким уровнем надежности и доступности. В процессе развертывания обычного сервера, как правило, многие сталкиваются со стандартной проблемой, когда запускаются и устанавливаются различные сервисы и службы, многие из которых не используются или используются редко. В процессе работы конфигурация многих таких служб теряется или забывается, и многие тайны настройки бывают утрачены (вроде как настроил и забыл). Поэтому при переезде на новый сервер всегда что-либо теряется (иногда что-то очень важное — например, почтовый ящик начальника). VServer решает эту проблему: вы делаете установку только один раз, а в дальнейшем работаете только с каталогом этого сервера, который можно легко скопировать и перенести на любое новое место — от обычного раздела на жестком диске до LVM-страйпа на внешнем FC дисковом массиве».

Учитывая все выше изложенное, вывод напрашивается сам собой: VServer — наиболее удобный и предпочтительный метод установки и использования сервера на базе операционной системы Linux. |

Роман Химов

Udev в разрезе

Структура устаревшей `devfs`, с точки зрения разработчиков, слишком громоздкая, запутанная и не очень удобная. Поэтому вполне логично, что на смену этой технологии пришла новая — `udev`. Как она работает и что дает конечному пользователю — об этом вы узнаете в этой статье.

| Проблемы /dev и devfs |

Концепция `devfs`, динамически создающей файлы виртуальных устройств, для своего времени выглядела очень привлекательно, однако практическая реализация была не очень удачной и в конечном итоге завела эту технологию в тупик. Если вы заглянете в статичную `/dev`, то увидите там около 18 000 файлов (с подкаталогами) — ну и где же там, к примеру, флеш-драйв? А еще, вам не кажется, что это довольно много, у вас ведь нет 18 000 физических/логических устройств? Или, наоборот, мало, если вы хотите построить систему с 10 000 SCSI-дисков! Наверное, было бы хорошо, если бы в `/dev` находилось только то, что реально присутствует в системе. Действительно, это очень здорово, и это есть, например, в FreeBSD — файлы устройств создаются динамически, `/dev` становится вполне читаемой как для пользователя, так и для программ. Аналогичная подсистема — `devfs` — есть и в Linux, казалось бы, какие еще проблемы?!

Однако они есть. Во-первых, `devfs` реально утяжеляет и усложняет ядро Linux, а это не любят ни разработчики, ни пользователи. А как быть с тем, что ее давно никто не хочет поддерживать (кстати, за все время существования в ядре она пребывала либо в состоянии `EXPERIMENTAL` — 2.4.x, либо `OBSOLETE` — 2.6.x)? А это еще что такое: `devfs`, оказывается, не совместима с `LSB`, когда почти все (по крайней мере из тех, что на виду) дистрибутивы стараются обеспечить совместимость с этим стандартом! Идем дальше — вернемся к внешним USB-устройствам хранения данных: покуда у вас одно такое устройство, все очень просто, `/dev/sda*` дает нам все, что надо, но если у вас их два? Вас никогда не смущал тот факт, что приходится запоминать, в каком порядке вы подключали флеш-накопители и внешние винчестеры, чтобы знать, что есть `/dev/sda`, а что `/dev/sdb`, `/dev/sdc` и т. д.? А с принтерами путаться не приходилось? Плюс ко всему, чисто с точки зрения архитектуры, разве заниматься именованием устройств — это дело

ядра? Его дело как раз в номерах `major/minor`, эти термины вообще оттуда и идут, имена ему совершенно безразличны. И, кстати, зачем нам, пользователям, знать номера `major/minor`?

Страшная картина вокруг, казалось бы, простой вещи — названий устройств, не так ли? Не совсем, конечно, так. Например, для серверов это, в общем-то, не играет никакой роли — конфигурация там одна и меняется крайне редко, там главное — чтобы действительно работало. Но мы-то с вами обычные пользователи настольного Linux и хотим простых и понятных удобств. Неужели все эти проблемы не разрешимы?

| Решение для /dev |

Давайте сначала посмотрим на еще одно новшество ядер 2.6.x — файловую систему `sysfs`. Туда экспортируется не просто масса, а прямо-таки тонны информации об устройствах — расположение на шинах, атрибуты вроде названий, классов устройств, серийных номеров и т. д., а также номера `major` и `minor`. То есть, оказывается, об устройствах мы легко все можем узнать и из пользовательского пространства! Теперь вспомним еще одну интересную вещь — механизм `hotplug`, которым ядра оснащаются еще начиная с ветки 2.4.x. В `/proc/sys/kernel/hotplug` можно посмотреть/изменить путь к одной маленькой, но очень важной программе, как минимум для настольной Linux-машины. Эта программа будет получать от ядра события, которые как раз касаются изменений в аппаратных конфигурациях. Тем самым мы и приходим к тому, что на самом деле вся информация, необходимая и даже избыточная для формирования `/dev`, уже давным-давно есть в пользовательском пространстве! Зачем тогда терзать ядро невнятными подсистемами вроде `devfs`?

Действительно незачем, знакомьтесь, наш герой — `udev`. Он органично сочетает возможности вышеописанных особенностей ядра Linux и дает нам то, что мы так хотели, — динамичный `/dev`! Кстати, а как он это делает?

Как это работает

Давайте проследим путь с самого начала — от ядра. Ядро видит появление нового устройства и запускает то, что мы прописали ему в `/proc/sys/kernel/hotplug`, передавая через переменные окружения все, что требуется `hotplug` для успешного выполнения своих функций. Это `$ACTION`, показывающий, что же случилось: добавили, удалили устройство или что-либо еще; `$SUBSYSTEM`, несущий в себе информацию о том, какая подсистема сгенерировала событие; `$DEVPATH`, содержащий очень важную вещь — путь к каталогу устройства в `/sys`; интересный параметр `$SEQNUM` — номер события, позволяющий упорядочить их и обрабатывать по очереди (события же могут доставляться непоследовательно), а также некоторые другие параметры, специфичные уже для конкретных подсистем.

Дело `hotplug` — загрузить нужные модули устройств, выполнить, возможно, их настройку и сообщить о событии `udev`. Сам по себе `hotplug` сегодня выполнен как солидный и весьма запутанный набор `shell`-скриптов. Однако Грег Кроа-Хартман, его автор, уже работает на `hotplug-ng`, выполненном на `C`, поскольку, хоть это и нормальный объем для `shell`-скриптов, он все же затормаживает загрузку системы, когда «симулянт событий» `coldplug` генерирует массу псевдособытий, но это к слову.

Вернемся к `udev`, застав его в момент получения события от `hotplug`. Стоп, какой такой `udev` получил событие от `hotplug` и как? Сам по себе `udev` тоже раскладывается на составные части в виде приложений `udevsend` и `udev`, а также демона `udev`. Зачем потребовался демон? Вспомним, события могут приходить в `hotplug` не в том порядке, в каком они реально происходили, — это явно требует кого-то, кто бы следил за порядком событий, поскольку, если событие `remove` придет перед `add`, наверное, что-то здесь не так, но что с этим делать `udev`? Вот и получается, что `udev` упорядочивает приходящие запросы так, чтобы они обрабатывались строго по мере появления.

Кто-то уже наверняка догадался, что `udevsend` здесь является связующим звеном, именно его вызывает `hotplug` (передавая параметры через переменные окружения), а `udevsend` уже знает, как доставить событие `udev`. Зачем же тогда приложение `udev`? Все дело в том, что непосредственную работу по созданию файлов устройств выполняет именно оно, `udev` не занимается этим напрямую — его дело следить за порядком. Вот так, через `hotplug`, `udevsend`, `udev` и `udev`, мы приходим к самому натуральному динамическому `/dev`!

Ну и что, разве мы не видели этого в `devfs`, к тому же без лишних приложений/демонов? Да, но не в таком виде. Одна из самых впечатляющих возможностей данного подхода — реализация абсолютно любой схемы именования устройств, ведь теперь она никак не навязывается нам ядром, и мы знаем все о наших устройствах! Хотите назвать ваш флеш-драйв `/dev/my-dear-flash` — пожалуйста! Все это потому, что Грег Кроа-Хартман, который и реализовал тот `udev`, что мы знаем (а ведь возможны и другие реализации, например его можно тесно совместить с `hotplug` в какой-нибудь маленький бинарный файл на `C` — это подойдет для встраиваемых систем), делал его с прицелом на настольные применения и максимальную гибкость в конфигурации. А что у нас с конфигурацией? Давайте посмотрим.

Конфигурация

`Udev` имеет основной конфигурационный файл `/etc/udev/udev.conf`, в котором задается расположение каталога устройств `udev_root` (а ведь можно попробовать сделать так, чтобы это был не `/dev`!); местоположение базы данных `udev` (`udev_db`); каталог с правилами `udev` (`udev_rules`); каталог с файлами, описывающими права пользователей на устройства (`udev_permissions`), права по умолчанию (`default_mode`), а также маленький параметр, включающий/выключающий логирование событий `udev` (`udev_log`).

Все, что касается интересных моментов `udev`, вынесено в каталог правил. Это стандартный `/etc/udev/rules.d`, хотя встречаются вариации. А там чаще всего вы обнаружите один файл с интересным названием `50-udev.rules`. Наверняка кто-то уже понял, что `50` — это неспроста, и будет прав: `udev` обрабатывает файлы правил (которые, кстати, должны обязательно оканчиваться на `.rules`) в лексикографическом порядке, то есть в том, который для нас вполне привычен и, как только находит подходящее правило для произошедшего события, останавливается. Соответственно, вряд ли стоит запускать руки в стандартный файл правил, можно просто создать какой-нибудь `10-vasya.rules` и описывать свои правила там: меньше вероятность испортить стандартные правила.

Файл состоит из ключей, а ключи разделяются запятыми. С помощью идентификационных ключей мы выделяем устройства, обязанные подчиняться этому правилу, а с помощью назначаемых указываем, что с ними надо делать в `/dev`; любое правило должно иметь как минимум один ключ каждого типа.

Ниже приводится список идентификационных ключей.

- ▶ **BUS** — шина, к которой подключено устройство, например `PCI`, `USB` или `SCSI`.
- ▶ **KERNEL** — имя устройства, данное ядром.
- ▶ **ID** — ID устройства на шине; например, для `PCI` это будет номер шины, а для `USB` — ID устройства.
- ▶ **PLACE** определяет место в топологии шины, например физический порт, в который включено `USB`-устройство.
- ▶ **SYSFS_имя-файла**, или **SYSFS{имя-файла}**, — проверка устройства по любому атрибуту из тех, что видны в `/sys`: это могут быть метки, вендоры, серийные номера, объемы и многое другое. Таких параметров в одном правиле можно задать до пяти штук.
- ▶ **PROGRAM** — еще одна мощная возможность — вызов внешней программы и проверка результата.

```
@kalis: /dev
srw----- 1 root root 4, 6 Июн 27 00:13 tty6
srw----- 1 root root 4, 5 Июн 27 00:13 tty5
srw----- 1 root root 4, 2 Июн 27 00:13 tty2
srw----- 1 root root 4, 1 Июн 27 00:13 tty1
srw----- 1 root root 4, 3 Июн 27 00:13 tty3
srw----- 1 root tty 7, 129 Июн 27 00:13 vcsa1
srw----- 1 root tty 7, 1 Июн 27 00:13 vcs1
drwxr-xr-x 2 root root 80 Июн 27 00:13 i2c-
srw----- 1 root root 89, 1 Июн 27 00:13 i2c-1
srw----- 1 root root 89, 0 Июн 27 00:13 i2c-0
lrwxrwxrwx 1 root root 16 Июн 27 00:13 cdrom -> cdroms/cdwriter1
drwxr-xr-x 2 root root 129 Июн 27 00:13 cdroms/
lrwxrwxrwx 1 root root 16 Июн 27 00:13 cdwriter -> cdroms/cdwriter1
lrwxrwxrwx 1 root root 16 Июн 27 00:13 dvd -> cdroms/cdwriter1
lrwxrwxrwx 1 root root 16 Июн 27 00:13 hdc -> cdroms/cdwriter1
lrwxrwxrwx 1 root root 16 Июн 27 00:13 hdd -> cdroms/cdwriter0
drwxr-xr-x 2 root root 1000 Июн 27 00:13 i2s
srw-rw---- 1 root uucp 4, 83 Июн 27 00:13 ttyS19
srw-rw---- 1 root uucp 4, 82 Июн 27 00:13 ttyS18
srw-rw---- 1 root uucp 4, 81 Июн 27 00:13 ttyS17
srw-rw---- 1 root uucp 4, 80 Июн 27 00:13 ttyS16
srw-rw---- 1 root uucp 4, 79 Июн 27 00:13 ttyS15
srw-rw---- 1 root uucp 4, 78 Июн 27 00:13 ttyS14
srw-rw---- 1 root uucp 4, 76 Июн 27 00:13 ttyS12
```

Часть структуры `/dev`

► **RESULT**: как раз через этот ключ можно сверять результат (возвращенную строку) вызова программы. Он должен следовать за вызовом **PROGRAM**.

Для успешного срабатывания правила необходимо, чтобы совпали все идентификационные ключи для рассматриваемого устройства.

Ключи целей:

- **NAME**: задает имя создаваемого файла устройства.
- **SYMLINK**: создает символическую ссылку на созданный файл устройства.
- **OWNER, GROUP, MODE** — стандартные владелец, группа и режим доступа к устройству соответственно.

В правилах можно применять подстановки в стиле `printf`:

- **%n** — номер, который дается устройству ядром; к примеру, для «sda3» это будет 3;
- **%k** — предлагаемое ядром имя устройства (в примере, указанном выше — «sda3»);
- **%M** — мажор-номер устройства;
- **%m** — минор-номер устройства;
- **%b** — ID устройства на шине;
- **%c** — результат выполнения программы, заданной через ключ **PROGRAM**; можно выбрать определенную часть этого результата, если они разделены пробелами, это делается через **%c{N}**, где **N** — номер части, а с помощью **%c{N+}** можно выбрать содержимое строки от начала **N**-го слова до конца строки;
- **%s{имя-файла}** — содержимое атрибута из `/sys`;
- **%e** — выдает минимальное целое **N**, если устройство с таким именем уже существует, после чего результирующее имя уже не будет совпадать с другим устройством; полезно при расстановке символических ссылок с целью обеспечения совместимости, а также нумерации устройств одного типа из разных подсистем ядра;
- **%%** — символ «%».

Кроме того, различные поля поддерживают шаблоны имен в стиле `shell`, то есть стандартные «*», «?» и «[]». Не многовато ли? Нормально, и сейчас вы в этом убедитесь сами.

Итак, посмотрим на простейшее правило `udev`:

```
KERNEL="fd[0-9]*", NAME="floppy/%n", SYMLINK="%k"
```

Это тривиальное правило работает для флоппи-дисков, которым ядро как раз и предлагает имя `fd{номер}`, оно создает реальный файл устройства в каталоге `(/dev/)floppy/` с простым числовым именем, а также символическую ссылку на него в виде привычного `fd{номер}`. Скучно? Конечно же, но давайте посмотрим дальше:

```
KERNEL="[hsd[a-z]", PROGRAM="name_cdrom.pl %M %m",  
NAME="%c{1}", SYMLINK="cdrom"
```

Этот пример приводил сам Грег Кроа-Хартман, самое большое его достоинство в скрипте `name_cdrom.pl`, который сперва определяет, является ли то устройство, которое ему дали, `CD-ROM`, а потом запрашивает по сети базу данных `CDDb` в поисках соответствующих аудиодиску исполнителя и названия альбома, после чего возвращает все это, а мы создаем файл устройства с таким именем (исполнитель-название) и ссылку `cdrom` на него! Впечатляет, не так ли? Этот маленький скрипт входит в стандартную поставку `udev` — можете посмотреть сами. А вот еще один пример, уже более близкий к жизни:

```
SYSFS{vendor}="TOSHIBA", SYSFS{model}="MK6021GAS",  
NAME="toshiba-drive"
```

Подобным образом представляется в `/sys` маленький внешний USB-винчестер 2,5", этим правилом мы его отлавливаем и даем имя «toshiba-drive». Что интересно — хотя у нас не было возможности проверить, но это же правило должно, по идее, отловить этот винчестер даже тогда, когда мы его подключим по FireWire — совершенно другой шине! После этого, вполне естественно, можно прописать где-нибудь в `fstab` простое и удобное для себя место монтирования этого конкретного диска — он уже не потеряется среди множества других.

Маленькое замечание, прежде чем вы начнете наводить порядок в своем `/dev`: для того чтобы изменения в правилах вступили в силу, необходимо запустить программу `udevstart`, которая заполняет `/dev` с нуля теми устройствами, которые найдет в `/sys`. Еще одна интересная утилита-компаньон `/dev`, о которой хотелось бы упомянуть, — `udevinfo`. Она позволяет получить различную информацию из базы данных `udev` и `sysfs`. А протестировать правила `udev` вы сможете с помощью утилиты `udevtest`.

Не только динамический /dev

Ну что ж, мы получили то, что хотели? Несомненно. Но `udev` приносит еще одно изменение — `/etc/dev.d/`. Там хранятся ссылки на те программы, которым `udev` будет передавать управление уже после того, как создаст/удалит файл устройства. То есть, цепочка продолжает развиваться, и мы можем аккуратно закриптовать что-нибудь, например, насчет конфигурации сети, после того как у нас появится или будет удалено какое-нибудь беспроводное устройство, или монтируется диск, для которого мы уже так заботливо определили однозначное имя в `/dev`. Именно через помощника в `dev.d` мы можем известить демона `HAL`, который дальше уже сможет рассказать в деталях о событии другим приложениям вашей свободной настольной среды. Имеет `dev.d` также и свою структуру, но за этим лучше обратиться к оригинальной документации `udev` на английском языке, которая входит в состав пакета `udev`, расположенного на сайте www.kernel.org/pub/linux/utils/kernel/hotplug/. Смело качайте свежую версию (а заодно обновите `hotplug`) и заглядывайте в каталог `docs`. Впрочем, возможно, вы предпочитаете инструменты своего дистрибутива.

Теперь, наверное, ни у кого не вызывает удивления то, что `udev` был очень хорошо принят разработчиками дистрибутивов — переход на него состоялся быстро даже среди тех, кто пытался использовать `devfs`, поскольку количество успешно преодолеваемых им проблем и качество их решения просто великолепны. Кстати, что касается `devfs`, то все тот же Грег Кроа-Хартман уже смастерил патчи, удаляющие ее из ядра. К тому моменту, когда вы будете читать эти строки, `devfs`, скорее всего, уже не будет являться частью ядра. Кстати, как уже говорилось, нам не нужны `major/minor`, одним из предложений Грега для Linux 2.7 (если эта ветка будет открыта) является последовательное или даже случайное выделение этих номеров для устройств — это проще и эффективнее, ведь теперь у нас есть инструмент, который уже готов с этим работать: свободный десктоп становится все лучше и удобнее. |

Анютинны грезы

Новое средство разработки с романтическим названием быстро набирает популярность. Работать с такой программой приятно вдвойне: она создана из любви к свободе и свободная по любви.

| Что в имени тебе ее |

Название сразу привлекает русского человека — нечасто встретишь программу, носящую русское и к тому женское имя, да еще в уменьшительно-ласкательной форме. Известно, что автор этой среды разработки живет в Индии и что программу он назвал в честь своей девушки (откуда она родом, Интернет умалчивает). На сегодня это все, что можно найти в разделе FAQ на сайте www.anjuta.org. А когда-то там же на первой странице можно было прочесть романтическую историю о том, с чего все начиналось: они не смогли встретиться в день ее рождения, молодой человек очень переживал из-за этого и решил написать среду разработки для GNOME (тогда еще версии 1.4), назвав ее именем любимой. Девушка была потрясена, хотя потребовалось время, чтобы объяснить ей, о чем речь, и осенью прошлого года они поженились. Такая вот красивая сказка наяву. Теоретически вскоре после свадьбы должна была выйти новая мажорная версия среды разработки, но реальность внесла свои коррективы, и Anjuta 2 увидела свет лишь весной этого года — и лучше бы разработчики подождали еще несколько месяцев.

| Межсезонье |

На сегодня есть две версии Anjuta — 1.2 и 2.0. Версия 1.2 стабильна, ее недостатки более или менее известны, локализованы (переведены на русский) и прикрыты фиговыми листочками там, где нужно. Версия 2.0 — новая, красивая, перспективная, но имеет кучу ошибок и недоработок, а некоторые огрехи видны практически невооруженным глазом.

Основной задачей нового проекта стало улучшение программного кода и тщательное проектирование ядра среды

разработки (чтобы через три года ее не пришлось переписывать снова). Была использована популярная идея вынести все, что только можно, в подключаемые модули. Подключаемым (формально) стал даже редактор. Много из того, что было в старой версии, попало в новую без существенных изменений. Вообще, с точки зрения функциональности новая версия не слишком отличается от старой. Поэтому сначала мы расскажем о старой версии, тем более что Anjuta 2 пока еще не совсем доведена до ума.

| То, что есть |

Пожалуй, сразу стоит сказать, что Anjuta 1.2 — это полноценная IDE, в которой, по крайней мере формально, есть все ожидаемые компоненты: навигация по проекту, CVS, отладчик. Рамки статьи не позволяют подробно остановиться на каждой части IDE, да это и не нужно: интерфейс в целом традиционный и не удивляет пользователя. Поэтому в основном мы будем говорить о специфике и чертах, характерных именно для Anjuta.

Среда разработки целенаправленно писалась для программирования под GTK+ и GNOME. Из этого следуют как библиотечные зависимости самой «Анюты», так и ее инструментарий. В частности, Anjuta 1.2 ориентирована на Autoconf/Automake (проекты на обычных Makefile тоже поддерживаются, но, разумеется, настройки проектов Automake к ним не применимы), есть мастера/друиды для соответствующих проектов (GNOME, GTK+). Для создания графического интерфейса предлагается, естественно, Glade, а для помощи по библиотечным функциям используется DevHelp, основанный на GtkDoc.

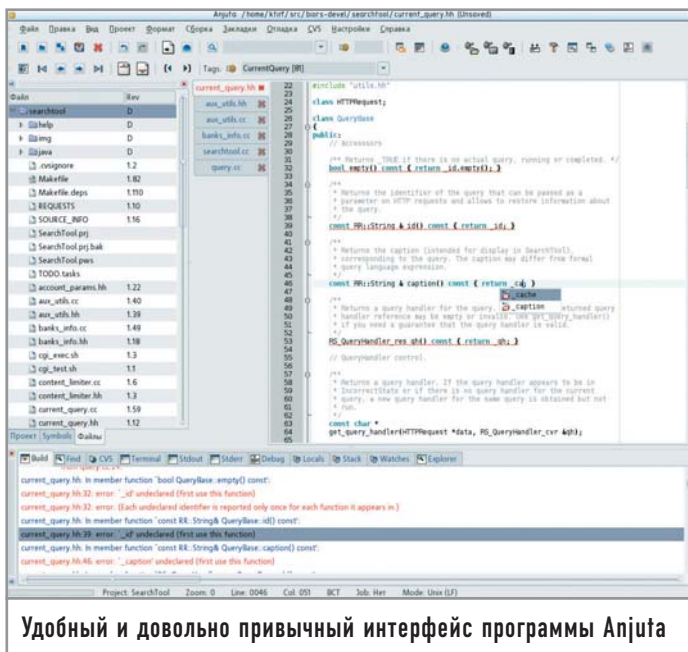
Другая характерная особенность Anjuta 1.2 — нетребовательность к ресурсам машины. Сам Наба Кумар (тот самый молодой человек, ныне счастливый муж) описывал Anjuta 1.2 как оболочку для уже существующих различных инструментов (по большей части консольных). Конечно, по системным требованиям это не KWrite или GEdit, но все же и не монстры типа Eclipse или KDevelop, которые могут потребовать 150 Мбайт виртуальной памяти.

Среда разработки в целом толерантна к различным языкам программирования; выражается это в подсветке синтаксиса для довольно большого набора языков, умении запускать нужный компилятор/интерпретатор для каждого файла и автодополнении, работающем беспорядочно (наподобие «Ctrl+N» и «Ctrl+P» в Vim), зато везде. А вот средства навигации по проекту не рассчитаны на что-либо отличное от C/C++.

Самая ценная вещь в Anjuta 1.2, которая, к счастью, осталась и в новой версии, — встраивание в среду внешних инструментов (пункт «Tools» в меню «Options»). Мало того что вы можете передавать значения различных переменных, связанных с текущим документом, выделением в файле, проектом и т. д., как аргументами, так и через стандартный ввод; можно еще указать, в каком меню будет находиться пункт для данного инструмента, какой у этого пункта будет значок и какой комбинацией клавиш его вызывать. Стандартный вывод и stderr внешней программы можно также перенаправить не только в окно сообщений, но и, например, заменить им текущее выделение в открытом файле. Нетрудно создать впечатление, что инструмент компилирован в «Анюту», а не подключен впоследствии — прекрасная реализация пресловутого Unix-way.

| То, что будет |

Что же может нам предложить новая версия по сравнению со старой? Во-первых, более гибким и мощным стал интерфейс среды разработки. Причем к улучшенному графическому до-



бавился богатый программный интерфейс, который, по замыслу разработчиков, и составляет главную ценность. Основная функциональность новой «Анюты» разбросана по различным модулям-плагинам, а ядро среды разработки — каркас, в который эти модули вставляются.

Уже сейчас модулей довольно много. Впрочем, пока они все написаны самими разработчиками и по большей части самоочевидны (например, Project Manager или редактор). В Anjuta 2 имеется поддержка Subversion (конечно, в виде модуля), которой не было в старой версии среды; модуль для CVS тоже есть. По разным меню пункты распределять больше нельзя, зато появились новые возможности по перенаправлению стандартных потоков. Довольно интересным может стать, по крайней мере со временем, модуль работы со списком «TODO» на базе gTodo (к сожалению, на данный момент пока еще нет автоматического добавления задачи в список, если в коде написать слово «TODO»).

К сожалению, весьма достойный список фиц — это не все. Графический интерфейс к сборке проектов теперь обеспечивается отдельным проектом gnome-build, находящимся на ранней стадии развития и поддерживающим пока лишь проекты Automake. В результате толерантности к языкам как не бывало — от нее осталась только подсветка синтаксиса (она обеспечивается редактором). Хуже того, даже проекты Automake вам не удастся настроить: окно свойств для них к выходу версии 2.0 так и не реализовано. Плагин для Glade находится в полуразобранном состоянии. Не сохраняются расположение панелей и подключенные плагины. Встраиваемый терминал приводит к краху всей «Анюты» почти при любом изменении его размеров (справедливости ради стоит сказать, что это одно из немногих действий, обрушивающих среду разработки). Хочется верить, что где-нибудь через полгода новая «Анюта» дорастет до того, чтобы ей можно было пользоваться в повседневной деятельности. Но пока этот продукт еще слишком сырой.

| Итоги |

Если вам нужна простая среда разработки без каких-то изысков, попробуйте Anjuta 1.2. Основные ее достоинства — низкие системные требования, простой интерфейс, мощный механизм подключения внешних программ. Старая версия неплохо подходит и для проектов «из одного файла», и для достаточно крупных разработок. Лучше всего Anjuta 1.2 проявляет себя в проектах GTK и GNOME среднего размера.

Что касается новой версии, то на данный момент мы не рекомендуем ее использовать, если только вы не собираетесь принимать участие в ее разработке. |

Полное затмение

В начале лета известная компания Macromedia объявила о своих планах присоединиться к Eclipse Foundation и разработке собственной версии IDE на базе Eclipse — Zorn. Список этой организации насчитывает около 100 членов, среди которых встречаются весьма громкие имена. Попробуем разобраться, чем же так привлекательна эта платформа.

| Немного истории |

Eclipse — это попытка создать некую обобщенную среду разработки. Проект зародился в недрах компании IBM, которая в 2001 году открыла его исходный код для широкой публики, одновременно создав организацию Eclipse.org. В разное время нее вступали Borland, QNX, Red Hat, Fujitsu, Oracle, SAP, Ericsson и многие другие компании. Постепенно проект становился все более независимым от IBM, а в прошлом году стал «самостоятельной» некоммерческой организацией (несмотря на то, что активным центром разработки Eclipse по-прежнему остается IBM).

В основе Eclipse лежит так называемая платформа (Eclipse Platform). Сама по себе она практически бесполезна и предоставляет лишь необходимый каркас. Остальные проекты работают поверх нее и являются различными средствами разработки и дополнениями (плагины) к ним и самой платформе. Среди них выделяются два основных проекта, номера версий которых совпадают с номерами версий платформы: это средства разработки на Java (Java Development Tools, JDT) и специализированная среда для разработки программ под платформу Eclipse (Plugin Development Environment). Кроме этого имеются средства разработки на C/C++ (C/C++ Development Tools, CDT), редактор UML-диаграмм, строи-

тель графических интерфейсов (для Java). В процессе создания находятся средства для веб-программирования (Web Tools) и разработки на Python, а также некоторые другие проекты. Помимо eclipse.org есть сайты <http://eclipseplugincentral.com> и <http://eclipse-plugins.info>, содержащие информацию о многочисленных плагинах для Eclipse.

| Внешний вид |

Основная часть кода Eclipse написана на Java. Пользовательский интерфейс построен на основе библиотеки SWT (Standard Widget Toolkit), сочетающей Java и C++. Стоит отметить, что это не самостоятельная графическая библиотека, а кросс-платформенная оболочка для графических библиотек. Под Linux SWT использует библиотеку Gtk+; внешний вид Eclipse при этом соответствует выбранной теме Gtk+. Использование C++ в критичных к производительности местах позволило избежать медлительности, которой обычно грешат программы, написанные на Java. Увы, объем занимаемой Eclipse оперативной памяти сопоставим с крупным Java-приложением: если у вас меньше 512 Мбайт, работать будет не слишком комфортно.

Интерфейс Eclipse простым не назовешь. В чем-то это следствие обобщенности платформы, в чем-то сказывается принад-

лежность к IBM. Немного облегчает жизнь хорошая документация к JDT, в том числе введение в работу с Eclipse. Есть документация и для средств разработки на C/C++, но она заметно хуже. К сожалению, и сам интерфейс Eclipse, и документация на русский язык не переведены. Желающие могут заняться переводом, но учтите, что объем строк для перевода только CDT – 600 кбайт, то есть примерно 400 страниц текста.

Eclipse – одна из самых функционально насыщенных сред разработки, как закрытых, так и открытых. По этой причине мы очень кратко остановимся на важных, на наш взгляд, моментах в работе с этой средой, а детальное исследование ее возможностей оставим читателю.

Для новичка самым непривычным в интерфейсе Eclipse будет, пожалуй, понятие перспективы (perspective). Она, в свою очередь, представляет собой набор и расположение панелей интерфейса – «видов» (views). Например, существуют отдельные перспективы для разработки на Java и C++, очень похожие друг на друга (вы можете вести разработку на двух языках в рамках одного проекта, по необходимости переключаясь между перспективами). Есть перспектива для отладки, в которой набор видов сильно отличается от аналогичного набора для разработки, а также перспектива для работы с системой управления версиями и другие.

Две перспективы могут не иметь абсолютно ничего общего между собой кроме открытых окон редактора. Открытые окна не меняются при смене перспективы. Иногда, как в случае переключения между разработкой и отладкой, это может быть весьма уместно. В иных случаях, например при переходе из C++ в Java, остающиеся открытыми окна с текстом на C++ могут раздражать.

Работа с проектами

Несмотря на обобщенность Eclipse, языков, на которых в этой среде можно нормально вести разработку, всего три: это Java, C и C++ (как вы уже поняли, Java поддерживается зна-

чительно лучше всех остальных). Мы постараемся побольше рассказать о C и C++, так как это самые распространенные языки в мире Open Source.

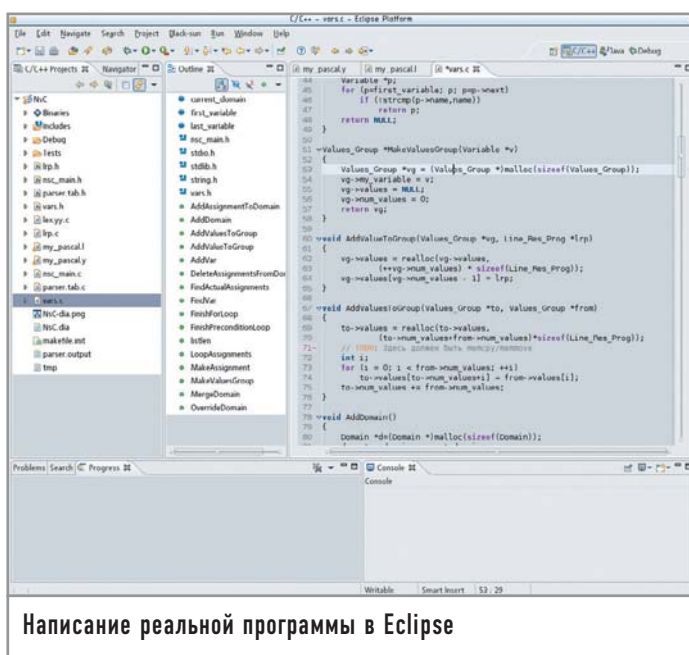
Поэтому начнем с CDT, то есть C/C++ Development Tools. Для C и C++ существует по два типа проектов – Standard Make и Managed Make. Если вы выбрали Standard Make, то Makefile будете писать сами, а среда лишь запустит Make, когда понадобится (вместо Make можно указать и другую программу). В проекте типа Managed Make среда берет на себя хлопоты по созданию Makefile, но и в этом случае остается возможность доработки правил сборки вручную. Eclipse различает C и C++, особенно это касается проектов Managed Make, поэтому следите за тем, из какого языка вы выбираете тип проекта.

В CDT полностью отсутствуют средства создания графических интерфейсов. Можно использовать любимый Glade или QtDesigner, но интеграции построителя GUI в среду разработки, увы, нет никакой. Нет в Eclipse и явной поддержки GNU Autotools, хотя это, в общем-то, и не мешает работать над использующими их проектами.

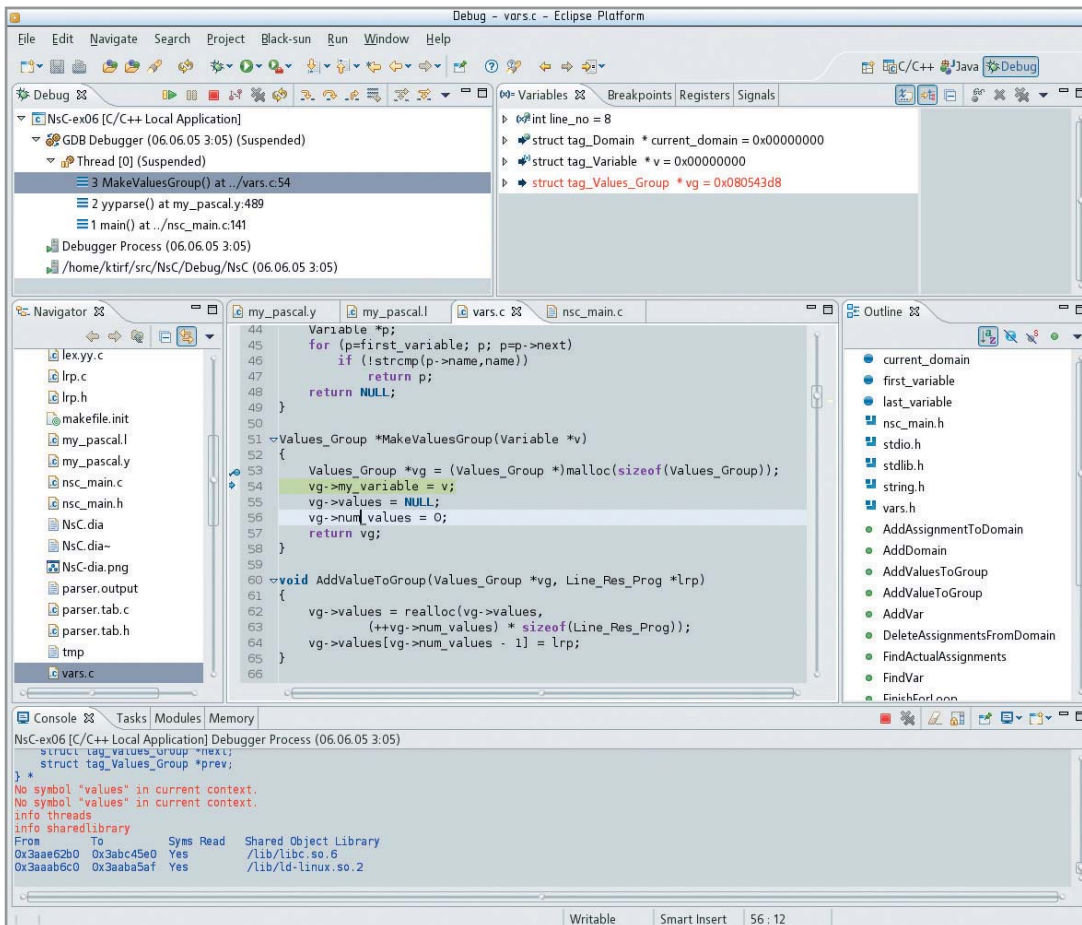
В остальном поддержка C и C++ реализована на уровне. Eclipse знает о шаблонах и пространствах имен C++, корректно работает с заголовочными файлами стандартной библиотеки, не имеющими расширения. Есть так называемые индексаторы (indexers), которые по своему назначению примерно соответствуют CTags (собственно, один из индексаторов в новой версии CDT как раз использует CTags). Элементы навигации показываются в двух списках – C/C++ Projects (с группировкой по проектам и файлам проектов) и Outline (только то, что в текущем файле). Немного неудобно то, что нет общего списка. С навигацией по проекту все в порядке; для быстрого перехода можно использовать как клавиатуру, так и контекстное меню, причем искать можно как объявление/определение, так и применения в тексте программы. В число элементов навигации входят и макросы.

Перейдем к Java Development Tools (JDT). Здесь всего один тип проекта, для сборки которого используется Ant. Редактировать правила Ant вручную, как и в проекте Managed Make из CDT, можно, но обычно не нужно. Построитель графического интерфейса под названием Visual Editor, привязанный к JDT, можно скачать с сайта обновлений и дополнений (update.eclipse.org) в виде плагина, после чего начать конструировать графические интерфейсы, использующие библиотеку SWT. В JDT вместо индексатора используются встроенные средства виртуальной машины, позволяющие получать списки пакетов/классов/членов. К началу июня в JDT уже появилась достаточно внятная поддержка обобщенных типов (generics) и некоторых других возможностей из J2SE 5.0.

В Eclipse есть отдельная перспектива, не зависящая от языка программирования, для работы с CVS. Для набирающей популярность системы управления версиями Subversion аналогичная перспектива имеется в составе плагина, который можно скачать с сайта разработчиков Subversion. Здесь поддерживается максимум возможностей, включая аннотирование исходников, сравнение различных редакций файлов, манипуляции с тегами и прочее. Создавая новые проекты,



Написание реальной программы в Eclipse



Отладка и отлов ошибок

можно наполнять их кодом, выгружая исходные тексты с указанного CVS-репозитория.

Причем, для того чтобы знать, в каком состоянии относительно CVS находится тот или иной файл, в другую перспективу переключаться не нужно. В любой момент информация о CVS-статусе файла показывается в каком угодно навигаторе проектов в виде маленького значка в углу иконки самого файла. Точно таким же образом можно узнавать и о том, в каких файлах были ошибки и предупреждения при сборке или проходе индексатора.

Редктирование исходных текстов

Редакторы в CDT и JDT похожи, хотя избранность Java налицо и здесь. Возможно, редакторы — это одна из самых сильных частей Eclipse по сравнению с другими средами разработки. Кроме уже традиционных складок (folds), подсветки синтаксиса и автоотступов разработчики добавили две по-своему уникальные вещи. Первая из них — это пометка изменений рядом с номерами строк, причем, что действительно полезно, изменения могут вычисляться не только по сравнению с последним сохраненным текстом, но и с версией этого файла в CVS. Вторая же приятная особенность заключается в том, что рядом с вертикальной полосой прокрутки было выделено пустое место, поперечные полоски на котором указывают, в какой части файла индексатором и/или компилятором выдавались ошибки или предупреждения. Если щелкнуть мышью по поперечной полоске, курсор попадет непосредственно на проблемную строку.

Автоматического форматирования кода в CDT нет — только автоотступы. В JDT есть не только автоматическое форматирование, которое в версии 3.1 можно настроить вплоть до мелочей, но и весьма приличные возможности переработки кода. Да и в целом редактор JDT более проработан и содержит массу приятных и полезных опций, хотя, чтобы их использовать, нередко приходится запоминать многочисленные комбинации клавиш и копаться в огромном окне настроек.

Автодополнение и подсказки о параметрах (calltips) работают во всех трех языках. После автодополнения имени функции или метода автоматически добавятся скобки, между которыми будет помещен курсор. Еще более мощным средством помощи при наборе являются шаблоны кода. Используются они следующим образом. Например, всем программистам на C известно обычное употребление функции malloc:

```
type *var = (type *)malloc(n*sizeof(type));
```

При наличии подходящего шаблона эта строка получается так: набираем название шаблона (прямо в тексте), вызываем автодополнение (по умолчанию — «Ctrl+Space»), которое разворачивает название шаблона в сам шаблон строки и ставит курсор на будущее имя типа в начале строки. Затем набираем имя типа (причем это происходит одновременно в трех местах — взамен каждого из слов type), нажимаем «Tab», вводим имя переменной, еще раз «Tab», затем выражение, которое в нашем примере заменено на «n». Еще одно нажатие «Tab» переносит нас в конец строки, и можно писать текст дальше. При этом значения приведения типа и sizeof будут подставлены автоматически.

В Eclipse «из коробки» шаблона для malloc, правда, нет, зато есть шаблоны для основных синтаксических конструкций и кое-что на тему printf. Написать свои шаблоны проще простого, с их помощью можно быстро создавать функции и классы, не затрачивая время на перемещения и набор повторяющихся фрагментов.

В такой монументальной среде разработки как-то неловко отдельно говорить о подсветке синтаксиса. Однако с подсветкой синтаксиса в Eclipse все хорошо до тех пор, пока вы работаете с Java, C/C++ и файлами описания сборки (для Ant и Make). Подсветки даже самого обыкновенного HTML у вас не будет, если вы не поставите нужный плагин. Стабильные плагины, дающие подсветку синтаксиса для большого количества языков, пока отсутствуют. К счастью, для частных случаев наподобие тех же HTML/XML/CSS решение уже есть в виде плагина под названием Black-sun (его можно найти на сайте проекта). Этот плагин вдобавок улучшает подсветку синтаксиса для Makefile.

Еще одна слабая сторона редактора — это закладки (bookmarks). Несмотря на поддержку такой функции, пользоваться ею не слишком удобно. Эту проблему можно успешно решить, установив плагин Quickmarks.

| Сборка и отладка |

Начнем с того, что и в JDT, и в CDT можно найти синтаксические ошибки без запуска компиляции. Упрощенная проверка синтаксиса ведется индексатором, который в состоянии найти такие ошибки, как необъявленная переменная или неправильно написанное имя класса. JDT еще и поможет сразу исправить ошибку (например, объявить необъявленную переменную; для отсутствующего метода создать заглушку) посредством нажатия «Ctrl+I».

Для компиляции в CDT используется GCC; в проектах Standard Make можно задействовать и другой компилятор благодаря полной свободе в написании Makefile. В случае с JDT также можно выбрать виртуальную машину Java и, соответственно, компилятор.

Все ошибки и предупреждения компилятора собраны на отдельной панели, которая называется «Problems». Прямо из нее с помощью двойного клика мышью можно попасть в строку кода, на которую ссылается то или иное сообщение. Например, в JDT нажатие комбинации клавиш «Ctrl+I» работает в том случае, если для строки есть возможность автоматического исправления.

На заметку

Любопытный факт

Не совсем понятно, откуда появилось название Eclipse (в переводе с английского «затмение»). Но обратите внимание, что в eclipse.org никогда не входила и не входит компания Sun. Более то-

го, у Sun есть своя среда разработки для Java, появившаяся еще до Eclipse. Но разработчиков, использующих Eclipse, больше. Похоже, IBM действительно удалось затмить солнце.

Что касается отладки, то здесь CDT всю использует возможности отладчика GDB. Фактически CDT — это графическая оболочка для GDB, имеющая довольно широкие возможности, хотя визуализации расположения структур в памяти наподобие той, что сделана в другой известной графической оболочке для GDB — DDD, здесь нет. Это немало, но и не так уж много, к тому же все-таки хотелось бы иметь профилировщик и отладчик памяти (например, в виде обертки для valgrind). У JDT есть свой отладчик, использующий ту же перспективу, что и отладчик CDT; таким образом, присутствует возможность вести отладку кода на обоих языках практически в одинаковом интерфейсе.

| Интеграция со средой пользователя |

Ни одна среда разработки не может содержать всех средств, которые могут понадобиться программисту. Eclipse позволяет непосредственно из среды разработки запускать внешние программы для обработки кода или выполнения других сопутствующих действий. В аргументах для внешних программ можно указывать различные переменные, связанные с текущим файлом, проектом или Eclipse в целом. Один из примеров уже упоминался выше: поскольку нативной поддержки Autotools в CDT не предусмотрено, можно указать все необходимое в виде внешних приложений, а Eclipse обновит нужные виды, когда приложения завершатся.

Еще одна форма интеграции в пользовательскую среду состоит в том, что, поскольку Eclipse не в состоянии открыть, например, файлы редактора диаграмм Dia, было бы неплохо, чтобы этот редактор запускался автоматически при попытке открыть подобные файлы. Для этого придется немного поработать вручную, а именно в разделе «General» в настройках вписать нужные маски файлов и пути к открывающим их программам. Получать MIME-типы из среды пользователя Eclipse пока не научилась, и есть опасение, что вряд ли научится — кросс-платформенность дает о себе знать.

| Резюме |

Все-таки можно говорить о том, что на Eclipse неизгладимый отпечаток наложило IBM-происхождение. Она, как почтенный мейнфрейм IBM S/370, велика в размерах, довольно сложна в освоении и необычайно эффективна для решения ряда задач. Eclipse не слишком дружит с популярными в Linux инструментами, такими как Autotools, Glade, valgrind. Основная ориентация на Java тоже не способствует популярности в стане Open Source. Вероятно, дело в том, что главная целевая аудитория Eclipse — разработчики крупного коммерческого софта, а не энтузиасты. Но тогда вообще неясно, почему нет оболочки для valgrind.

В любом случае, если вы собираетесь писать консольные программы или демоны на C/C++ и не боитесь сложностей освоения среды разработки, обязательно попробуйте Eclipse: средства управления проектами, редактор исходного кода и отладчик в CDT — просто превосходные! Если же вы пишете программы на Java, то и так наверняка давно знаете, что Eclipse — самая популярная и одна из лучших сред разработки для этого языка. |

Рабочая лошадка администратора

Язык программирования Python в последнее время становится все более популярным. Он по умолчанию включается в большинство дистрибутивов Linux, его можно установить на FreeBSD (кстати говоря, Perl с некоторых пор также не является частью системы, так что и в этом плане он уравнивается в правах с Python). Многие сторонние программы наряду с поддержкой Perl обеспечивают и работу с Python (например, модуль `mod_python` к Apache, язык PL/pgsql в PostgreSQL).

Поднимать вечный спор «Perl vs Python» не будем: с обеих сторон можно привести массу доводов как «за», так и «против». На наш взгляд, здесь решающим фактором при выборе будет, скажем, психологическая совместимость языка и программиста. Например, некоторым Python пришелся по душе из-за строгого синтаксиса и очень удобной концепции повторного использования кода.

В данной статье мы рассмотрим несколько примеров использования Python для решения задач администрирования систем Unix (все примеры оттачивались на FreeBSD, но и на Linux должны работать не хуже). Предполагается, что большинство пользователей хотя бы в общих чертах знакомы с синтаксисом Python. Если вы не относитесь к их числу, переходите к следующему разделу, в котором коротко описаны основные моменты, необходимые для чтения представленного в статье кода.

Несколько вводных замечаний

Напомним основные моменты синтаксиса языка Python. Прежде всего, блоки кода в нем выделяются не операторными скобками, как в C или Perl, а отступами. Каждая команда записывается на одной строке (в принципе синтаксис позволяет объединять несколько команд в одну строку или переносить

команду на следующую; однако это ухудшает читаемость кода и настоятельно не рекомендуется). Все строки одного блока (цикла, подпрограммы, ветвления и т. д.) должны быть выровнены на одинаковое количество пробельных символов (пробелами или табуляцией). Например:

```
for file in dir:
    print file
    print '-----'
print file
```

В данном фрагменте телом цикла `for` являются первые две строки `print` (заданы с отступом относительно оператора `for`). Последняя строка будет исполнена после выхода из цикла, поскольку она уже не имеет отступа.

Как вы, должно быть, заметили, команды не нужно завершать символом «;» (он должен указываться только при размещении нескольких команд в одной строке). Поскольку перевод строки рассматривается как конец команды, для ее переноса символ конца строки должен экранироваться:

```
print a, b, c, d, e,
    f, g, h
```

Строку внутри кавычек можно переносить без экранирования, поскольку в этом случае перевод строки считается принад-

лежащим строковой константе (обязательно убедитесь, что этот символ не исказит отображение строки).

В Python идея повторного использования кода заложена в основу синтаксиса. Если вы когда-либо писали какую-то функцию (или получили готовую), то ее очень легко подключить к любому другому сценарию с помощью оператора import:

```
import sys
from Tkinter import *
```

В первой строке мы подключаем модуль sys, включенный в стандартную библиотеку Python. Во второй импортируются все функции модуля Tkinter. Причем в качестве модуля может рассматриваться любой файл, содержащий код Python. Отличие этих способов заключается в том, что при «простом» импорте для всех функций подгружаемого модуля создается свое пространство имен, и при обращении к той или иной функции (или переменной) следует указывать имя модуля (см. примеры ниже). А конструкция «from <module_name> import <list_of_functions>» добавляет импортируемые функции в пространство имен выполняемого сценария, что позволяет обращаться к функции без указания имени модуля. Естественно, импортируемое таким образом имя не должно конфликтовать с уже имеющимися. Кроме того, во втором случае вместо символа «*» можно явно перечислить функции, которые должны быть доступны в разрабатываемом сценарии.

Комментарии, как и в Perl, начинаются с символа «#».

Теперь рассмотрим несколько примеров решения различных задач администрирования.

| Автоматическое редактирование файлов |

| Исходные условия |

Имеется древовидная структура каталогов, содержащая текстовые файлы. Для определенности будем считать, что такой структурой является дерево веб-сайта некоторой компании (простые HTML-файлы). Пусть по ряду причин компания сменила название с Indians на Cowboys, что отразилось и на доменном имени сайта — вместо www.indians.com будет www.cowboys.ru.

| Задача |

Реализовать автоматическую корректировку всех HTML-файлов, заменив все названия и доменные имена новыми значениями. Будем считать, что регистр символов везде соблюдается безукоризненно, то есть нигде в тексте не встречается написание типа InDiAnS.

| Решение |

Сначала приведем полностью итоговый сценарий, учитывая его небольшой размер. Необходимые пояснения размещены ниже.

```
#!/usr/local/bin/python
import os, sys
from glob import glob
exts = ['html', 'htm']
old = ['Indians', 'indians.com']
new = ['Cowboys', 'cowboys.ru']
DEBUG = 1
```

```
def indir(curdir):
    global space
    dir = glob('*')
    for file in dir:
        ext = os.path.splitext(file)[1][1:]
        if os.path.isdir(file):
            if DEBUG: print '%s---> %s' % (' '*space, file)
            os.chdir(file)
            space += 1
            indir(file)
            space -= 1
            os.chdir('.')
            if DEBUG: print '%s<--- %s' % (' '*space, file)
        elif exts.__contains__(ext):
            try:
                text = orig = open(file, 'r').read()
                i = 0
                for word in old:
                    text = text.replace(old[i], new[i])
                    i += 1
                if orig != text:
                    open(file, 'w').write(text)
                    if DEBUG: print '%s+ %s...' % \
                        (' '*space, file)
                else:
                    if DEBUG: print '%s= %s...' % \
                        (' '*space, file)
            except:
                print 'I/O ERROR while %s processing' % file
```

```
def Usage():
    print 'Usage: fe.py startdir'
    print
    space = 0
    if len(sys.argv) != 2:
        Usage()
    else:
        startdir = sys.argv[1]
        if DEBUG: print 'Start with %s' % startdir
        os.chdir(startdir)
        indir(startdir)
```

Самая первая строка указывает операционной системе (если речь идет о Unix) путь к командному интерпретатору Python. Приведенный путь характерен для FreeBSD, в Linux это обычно /usr/bin/python. В Windows она воспринимается как обычный комментарий.

Здесь нам понадобились модули os и sys, а также функция «glob» из одноименного модуля. Все модули входят в стандартную поставку Python, искать их нигде не нужно. Задаем список обрабатываемых расширений (переменная exts) и два списка заменяемых значений — в первом старые строки, которые должны быть заменены, во втором — новые (оба списка должны строго соответствовать друг другу).

Установка значения «1» переменной DEBUG позволит получить в окне терминала подробные сообщения о процессе исполнения сценария.

Далее следует определение функции «indir» (оператор def). К ней мы вернемся немного позже, пока лишь отметим, что к ней принадлежат все строки, имеющие отступ.

Функция «Usage» выводит на экран сообщение о синтаксисе запуска сценария.

Последние строки сценария выполняют следующие действия: переменная `space` инициализируется значением «0» (она будет нужна в функции «indir»); проверяется количество переданных сценарию аргументов, и если оно не равно двум (имя самого сценария и один параметр), то вызывается функция «Usage». То, что передано в командной строке, можно получить из специального списка `argv` модуля `sys`. Переменная `sys.argv[0]` хранит имя запущенного сценария, поэтому первый аргумент имеет индекс 1. Если длина списка `sys.argv` равна двум, считывается аргумент, переданный в командной строке (мы будем передавать в сценарий имя каталога, являющегося корнем обрабатываемого дерева). Далее выводится отладочное сообщение; работа операторов `if` и `print`, как нам кажется, понятна без комментариев. Разве что следует обратить внимание на использование знакомест при выводе текстовой строки: в самой строке место, куда должно быть выведено значение строковой переменной, отмечается конструкцией `%s`, при обработке вместо нее подставляется конкретное значение соответствующей переменной — одной из перечисленных после символа «%».

Функция «`chdir`» модуля `os` делает текущим указанный каталог, а последней строкой вызывается функция «indir», которая и будет выполнять основную работу.

В функции «indir» мы сначала объявляем глобальную переменную. Без такого объявления `space` будет считаться локальной и не примет во внимание значение, установленное вне функции. Назначение переменной `space` рассмотрим несколькими абзацами ниже.

Функция «`glob`» модуля `glob` (поскольку она была импортирована в пространство имен нашего сценария, то при обращении к ней имя модуля не указывается) возвращает список имен файлов и каталогов, найденных в текущей папке и соответствующих указанному шаблону. В данном случае мы используем шаблон «*», поскольку имена подкаталогов могут быть любыми. Обратите внимание, что в Unix DOT-файлы (имена которых начинаются с точки, включая каталоги `.` и `..`) не соответствуют шаблону «*» (чтобы убедиться в этом, наберите «`echo *`» в командной строке), поэтому дополнительных мер по исключению заикливания на родительском каталоге предпринимать не требуется.

Таким образом, переменная `dir` теперь содержит список всех файлов и каталогов, находящихся в текущей папке.

Цикл `for` в Python позволяет простым и естественным образом проходить по всем элементам списка. В нашем случае тело цикла будет выполнено для каждого имени файла или каталога из `dir`. Текущий элемент заносится в переменную `file`.

Следующая строка, призванная дать нам расширение обрабатываемого файла, выглядит не совсем понятно:

```
ext = os.path.splitext(file)[1][1:]
```

Здесь функция «`splitext`», находящаяся в подмодуле `path` модуля `os`, разбивает полное имя файла, переданное как параметр,

на две части — все, что стоит до расширения (до последней точки), и само расширение. Значения возвращаются в виде списка, в котором нас интересует второй элемент. Рассматривая функцию `os.path.splitext(file)` как переменную-список (ее результат и будет занесен во временную переменную), мы применяем к ней обычную процедуру извлечения элемента по индексу. Таким образом, конструкция `os.path.splitext(file)[1]` будет представлять собой внутреннюю переменную, содержащую расширение файла. Однако в эту строку попадает и ведущая точка, то есть ее содержимое на данном этапе выглядит как `.html`, и эту точку следует убрать.

В Python символьная строка может рассматриваться как список букв (выполните в интерфейсе интерпретатора команду `for a in 'qwerty': print a`, чтобы убедиться в этом). Для списков существует особая операция — срез. Например, чтобы получить список, содержащий элементы с третьего по пятый некоторого существующего списка, можно выполнить следующее:

```
Sublist = list[3:6]
```

Таким образом, указывается диапазон индексов, причем элемент, имеющий индекс, указанный вторым, в результирующий список не включается (то есть `sublist` будет содержать элементы 3, 4 и 5). Если опущен первый из индексов, подразумевается «с начала списка», если опущен второй — «до конца списка». Теперь становится понятной и последняя запись разбираемой строки: в переменную `ext` записывается не вся строка расширения, а подстрока, начиная со второго символа и до конца. Как видите, интерпретатор Python позволяет осуществлять довольно сложную обработку без явного указания переменных.

Вернемся к рассмотрению функции «indir». Если очередной элемент из переменной `dir` представляет собой имя каталога (проверяется функцией «`isdir`» модуля `os.path`), то выполняется переход в этот каталог, рекурсивный вызов «indir» и возвращение в родительский каталог после завершения обработки.

Следует чуть более подробно остановиться на записи «`' '*space`», которая является переменной для заполнения первого знакоместа в отладочных сообщениях. Оператор «*» для строк означает «повторить указанное количество раз». Глобальная переменная `space` как раз эти «разы» и содержит, увеличиваясь при входе в подкаталог и уменьшаясь при возврате из него на один. В результате вывод на экран представлен «лесенкой», отражающей вложенность каталогов.

Если очередной элемент является регулярным файлом (конечно, есть еще и символьные ссылки, файлы-«дырки» различных устройств и т. д., но в данном примере мы про это «забудем»), то проверяем его расширение (часть `elif` оператора `if`). Здесь удобно использовать внутреннюю функцию «`__contains__`», которая применена к списку и возвращает «истину», если список содержит элемент, равный ее аргументу. Таким образом, дальнейшей обработке будут подвергаться только файлы, расширения которых упоминаются в списке `exts`, заданном в начале сценария.

Собственно обработка файла достаточно проста: его содержимое считывается в переменную `text` (копия — в переменную `orig`; Python допускает множественное присваивание), к которой в цикле применяется метод `replace` для каждой пары значе-

ний. Запись в файл выполняется только в том случае, если измененный текст (text) отличается от orig. Но поскольку операции с файлами небезопасны (может оказаться недостаточно прав на чтение или запись), их целесообразно выполнять внутри оператора «try .. except ..»: если при выполнении команд блока try возникнет ошибка, управление будет передано на блок except, и выполнение сценария продолжится.

Обратите также внимание, что при работе с файлом нигде не указывается его дескриптор. То есть мы используем тот же прием, что и ранее при отделении расширения от имени файла — неявные переменные. В классическом варианте чтение из файла выглядело бы следующим образом:

```
fd = open(file, 'r')
text = fd.read()
fd.close()
```

Кстати, поскольку неявная переменная-дескриптор существует только внутри данной команды, то при переходе на следующую файл будет автоматически закрыт, поэтому функцию «close()» вызывать не обязательно, чем мы и воспользовались.

На выводе рассмотренного сценария символом «+» будут отмечены файлы, в которых были сделаны корректировки, «=» — которые оставлены без изменений. Скорость работы оказалась достаточно приличной — в дереве общим объемом около 5 Мбайт (235 файлов, подлежащих обработке) тройная замена была выполнена за пару секунд.

| Анализ лог-файлов Apache |

| Исходные условия |

Имеется веб-сервер Apache, ведущий лог-файл (для определенности — /var/log/httpd-access.log) посещения страниц вида:

```
x.x.x.x — [16/Jun/2005:07:06:44 +0400] "GET /city/city.htm
HTTP/1.0" 304 — "-" "ConveraCrawler/0.8
(+http://www.authoritativeweb.com/crawl)"
y.y.y.y — [16/Jun/2005:07:33:11 +0400] "GET /map.html
HTTP/1.0" 200 3960 "-" "Googlebot/2.1
(+http://www.google.com/bot.html)"
```

| Задача |

Получить статистику посещений каждой из страниц, а также рассчитать общий объем отданных клиентам страниц.

| Решение |

Первая строка приведенного выше примера лог-файла соответствует ответу с кодом 304 (Not Modified) и интереса для нас не представляет (если, конечно, не потребуется собирать статистику и по таким ответам). В данном примере учитывать следует только ответы с кодом 200 (страница отдана клиенту).

Код сценария в данном случае заметно проще, но тоже имеет ряд особенностей:

```
#!/usr/local/bin/python
# -*- coding: koi8_r -*-
fd = open('/var/log/httpd-access.log', 'r')
stat = {}
cnts = {}
line = 'not empty'
```

```
while 1:
    line = fd.readline()
    if not line:
        break
    fields = line.split(' ')
    if fields[5] == "GET" and fields[8] == '200':
        url = fields[6]
        if url.find('?') != -1:
            url = url[:url.index('?')]
        traf = int(fields[9])
    else:
        continue
    if stat.has_key(url):
        stat[url] += traf
        cnts[url] += 1
    else:
        stat[url] = traf
        cnts[url] = 1
fd.close()
statkeys = stat.keys()
statkeys.sort()
totaltraf = 0
totalcnt = 0
print '-' * 60
print '%-25s %12s %12s' % ('Страница', 'Обращений', 'Тра-
фик')
print '-' * 60
for key in statkeys:
    print '%-25s %12d %12d' % (key, cnts[key], stat[key])
    totaltraf += stat[key]
    totalcnt += cnts[key]
print '-' * 60
print '%-25s %12d %12d' % ('Total:', totalcnt, totaltraf)
print '-' * 60
```

Вторая строка сценария указывает используемую кодировку, что необходимо, так как в данном примере используются символы, выходящие за рамки стандарта ASCII-128. Если кодировку не указать, то сценарий станет работать, но каждый раз будет выводиться предупреждение.

Итак, открываем файл, в цикле считываем его построчно посредством метода readline(). Как только будет прочитана пустая строка — выходим из цикла. Каждую строку разбиваем на составляющие, используя в качестве разделителя пробел (метод split). Если строка содержит на шестой позиции символы «GET», а на девятой — «200» (нужный нам код ответа), то записываем в переменные седьмое поле, содержащее URL запрошенной страницы, и десятое, в котором указано количество переданных байт.

При обработке строки URL используется уже знакомый нам срез, чтобы отбросить все, что находится после символа «?», то есть получить просто имя страницы без параметров, которые передаются на сервер. Позицию знака «?» возвращает функция «index», а чтобы не возникало ошибки при отсутствии его в строке, предварительно проверяем его наличие функцией «find».

Немного подробнее остановимся на переменных `stat` и `cnts`. Это ассоциативные массивы, хранящие пары «ключ-значение». В роли ключа будет выступать имя страницы, а в роли значения — трафик и количество обращений.

Метод `keys`, примененный к ассоциативному массиву, возвращает список ключей; метод `sort` осуществляет сортировку списка.

Далее в цикле по всем элементам в `statkeys` результат выводится на экран и попутно вычисляются суммарные значения числа обращений и трафика.

Обратите внимание на цифры в конструкциях `%s` и `%d` — они указывают число символов, которое будет занимать переменная. По умолчанию выравнивание осуществляется по правому краю с помощью пробелов. Минус перед числом заставляет осуществлять выравнивание влево.

По результатам работы будет выведена таблица с указанием количества обращений к каждой странице и объема трафика.

Обработка лог-файла размером 17,5 Мбайт на Celeron 466 (128 Мбайт памяти) занимает около 10 секунд.

| Сбор статистики работы канала Интернета |

| Задача |

Каждые 15 минут требуется проверять работоспособность канала путем отсылки десяти пакетов размером 1400 байт. Статистику (количество принятых пакетов, среднее время переноса и среднюю девиацию `stddev`) будем записывать в лог-файле, а при отсутствии соединения — отправлять письмо администратору.

| Решение |

Цель данного примера — показать приемы взаимодействия Python с операционной системой. Для проверки работы канала будет использоваться системная утилита `ping`, отправка уведомления будет осуществляться с помощью команды `mail`, а логи будут записываться с помощью программы `logger`.

Для взаимодействия с программами операционной системы используются две функции модуля `os` — «`popen`» и «`system`». Первая из них открывает канал с указанной в качестве аргумента программой, что позволяет считать ее вывод с помощью метода `readlines` (или любого другого, который лучше подходит в той или иной ситуации). Функция «`system`» просто выполняет указанную команду и восстанавливает лишь код возврата.

Код сценария представлен ниже:

```
#!/usr/local/bin/python
import os
pingcmd = '/sbin/ping -q -c 10 -s 1400 '
defaultthost = 'ya.ru'
loggercmd = '/usr/bin/logger'
mailcmd = '/usr/bin/mail -s \'Ping notification\'
admin = 'admin@site.ru'
def tolog(msg, host):
    os.system(loggercmd + ' "ping ' + host + ': ' + msg + '"')
def tomail(msg):
    os.system('echo ' + msg + '|' + mailcmd + ' ' + admin)
def pingchannel(host=defaultthost):
    reply = os.popen(pingcmd + host).readlines()
```

```
received = int(reply[3].split(' ')[3])
if received == 0:
    tolog('%2d | %7s | %7s' % (received, '-', '-'), host)
    tomail('No ping to %s' % host)
else:
    stat = reply[4].split(' ')[3].split('/')
    average = float(stat[1])
    stddev = float(stat[3])
    tolog('%2d | %7.3f | %7.3f' % (received, average, stddev),
host)
pingchannel()
pingchannel('samag.ru')
pingchannel('cam12.noping.ru')
```

Большинство используемых здесь команд применялись ранее. Поясним лишь конструкцию `host=defaultthost` в определении функции «`pingchannel`». Такая запись позволяет определить значение аргумента по умолчанию — если функция вызывается с аргументом (последние две строки), то применяется переданное значение. Если же аргумент при вызове функции опущен, то он принимает значение, указанное после знака равенства.

Подробности работы использованных в сценарии системных утилит (`mail`, `logger`, `ping`) смотрите на соответствующих страницах справочного руководства вашей операционной системы.

В результате работы рассмотренного сценария в системный журнал (согласно настройкам `syslog`) будут помещаться следующие строки:

```
Jun 17 16:07:06 host serg: ping ya.ru: 10 | 34.307 | 0.529
Jun 17 16:07:08 host serg: ping samag.ru: 9 | 11.772 | 0.058
Jun 17 16:07:21 host serg: ping cam12.noping.ru: 0 | — |
```

Если с какого-то из тестируемых узлов не будет получено ни одного ответа, администратору на электронный ящик будет отправлено сообщение об этом:

```
No ping to cam12.noping.ru
```

Данный сценарий можно периодически запускать по `cron`, и в итоге будет накапливаться статистика качества работы каналов. Лог-файл можно обрабатывать также сценарием на Python, основные принципы такой обработки были рассмотрены в примере 2.

| Заключение |

Приведенные выше примеры показывают, что Python может с успехом применяться для решения самого широкого круга задач администрирования. Он удобен для работы с файлами, хорошо обрабатывает текстовые документы (хотя в этом он несколько уступает мощи и гибкости Perl), предоставляет необходимые средства для взаимодействия с операционной системой.

Благодаря концепции повторного использования кода через некоторое время нарабатывается целая «библиотека» различных функций (например, функцию «`pingchannel`» из третьего примера можно будет в дальнейшем использовать и в других сценариях, нуждающихся в проверке работоспособности канала), и программировать становится все проще. А строгий синтаксис дает гарантию того, что через несколько лет вы сами (или пришедший вам на смену системный администратор) сможете без труда разобраться в вашем сценарии. |

Фактический материал

В любой даже самой отточенной системе всегда есть резерв для совершенствования. Этот раздел поможет вам сделать работу в Linux удобной и продуктивной, возможно, здесь вы найдете решение какой-либо программной или аппаратной проблемы. Начнем с советов по работе с графическими средами GNOME и KDE.

Обе среды довольно удобны, но, скорее всего, вам наверняка чего-то не хватает. Возможно, это какое-то действие в пункте меню, которое вы часто выполняете вручную и хотели бы автоматизировать этот процесс, или упаковка файла каким-то специфическим архиватором, или изменение прав доступа к нему.

KDE

Создание собственных действий в сервисном меню

Давайте создадим дополнительную команду, которая делает файл исполнимым. (Как говорится, был обычный файл, а стал программой.) Это действие может оказаться очень полезным, если вы часто пишете сценарии bash. Ведь сценарий bash — это обычный текстовый файл, созданный в текстовом редакторе. Чтобы он запускался, нужно сделать его исполнимым. Данный совет посвящается любителям все делать мышкой — тем, кому лень вводить команду вида «chmod +x имя_файла».

Итак, в любимом текстовом редакторе создайте файл следующего содержания:

```
[Desktop Entry]
```

```
ServiceTypes=all/allfiles
```

```
ServiceType=application/x-shellscrip
```

```
Actions=MakeExe
```

```
[Desktop Action MakeExe]
```

```
Name=Make executable
```

```
Name[ru]=Сделать файл исполнимым
```

```
Exec=chmod +x %f
```

```
Icon=kfm
```

Рассмотрим первую секцию. Первая опция задает тип файлов, для которых можно выполнить указанное действие. В данном случае оно доступно для всех файлов (allfiles). Если нужно выполнить какое-то действие для каталога, то параметр ServiceTypes должен принять значение inode/directory:

```
ServiceTypes=inode/directory
```

Вообще, в качестве значения данного параметра можно указать любой MIME-тип, например:

```
ServiceTypes=audio/x-mp3
```

Если требуется выполнить какое-то действие для всех типов файлов кроме каких-то определенных, используется параметр ExcludeServiceTypes. Например, если вы определяете действие архивирования, то должны указать сервисные типы — все файлы, но исключить архивы:

```
ServiceTypes=all/allfiles
```

```
ExcludeServiceTypes=application/x-zip,kdedevice/*
```

Параметр Actions определяет действия, описанные в файле. В данном случае описано только одно из них — MakeExe, определенное в секции [Desktop Action MakeExe]. Параметр Name — это надпись, которую вы найдете в сервисном меню KDE. Это общая надпись, которую увидит пользователь, ее желательно писать на английском языке. Параметр Name[ru] — это тоже надпись, описывающая действие, но на русском языке. Пользователь увидит ее, если у него KDE на русском языке.

Параметр Exec — это команда, которая будет выполнена, а %f — параметр, определяющий имя файла, по которому вы щелкнули правой кнопкой; нужное имя будет подставлено вместо %f.

Созданный файл сохраните под именем make_exe.desktop.

Как видите, в этом нет ничего сложного. Осталось только сохранить файл в нужном каталоге — servicemenus. Он находится в каталоге /usr/share/apps/conqueror/:

```
$ sudo cp make_exe.desktop
```

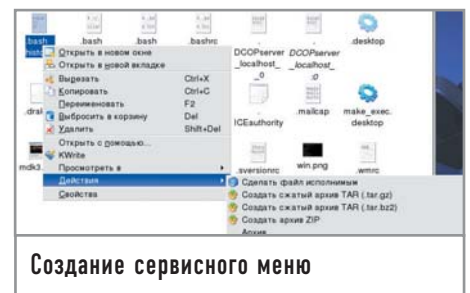
```
/usr/share/apps/conqueror/servicemenus/
```

Для записи в этот каталог нужны права суперпользователя, поэтому, чтобы выполнить команду cp, понадобится команда sudo, которая и обеспечит нужные полномочия (вам нужно будет ввести пароль пользователя root). Если же вы не администратор системы и хотите добавить команду в меню только локального пользователя, скопируйте файл make_exe.desktop в каталог ~/.kde/share/apps/conqueror/servicemenus/:

```
$ cp make_exe.desktop
```

```
~/.kde/share/apps/conqueror/servicemenus/
```

Результат вашей работы появится в параметре Actions сервисного меню KDE.



GNOME

Меню «Сценарии»

Меню «Сценарии» аналогично меню «Действия» в KDE. Идентичен и принцип работы. Мы определяем команды, которые будут выполнены при выборе того или иного пункта меню. Только если в случае с KDE файл меню имел определенный формат, то для GNOME файл

меню — это обычный bash-сценарий. Хорошо это или плохо, зависит от степени вашего знакомства с bash.

Если вы знаете его хорошо, то сможете создавать очень сложные сценарии, позволяющие автоматизировать огромное количество рутинной работы. Действия вашего сценария не будут ограничены форматом файла, как в KDE. Ведь, по сути, в KDE для выполнения действия мы можем определить только простые команды. Если нужно использовать то же разветвление (if ... then), все равно придется писать сценарий на bash, а из меню KDE вызывать не какую-то программу, а именно созданный вами сценарий.

Если же вы не очень хорошо знакомы с bash, рекомендуем его выучить, а пока вы будете это делать, лучше создавать дополнительные пункты меню в KDE.

Идея создания пункта меню достаточно проста. Вы пишете bash-сценарий и помещаете его в каталог ~/gnome2/nautilus-scripts/. В нем же можно создавать подкаталоги — они будут отображаться как дополнительные меню. На скриншоте видно, что в каталоге nautilus-scripts был создан подкаталог Convert, а в нем — файл Convert2Gif. Думаем, с этим все ясно. Теперь рассмотрим пример самого сценария:

```
#!/bin/bash
convertprg=`which convert`
while [ $# -gt 0 ] ; do
    picture=$1
    filetype=`file $picture | cut -d ' ' -f 3`
    if [ $filetype = "image" ]
    then
        newfile=`echo "$pic-
```

```
ture" | cut -d . -f 1`
        $convertprg "$picture" "$newfile".gif
    fi
done
```

Данный сценарий выполняет конвертирование изображения в формат GIF. В качестве программы-конвертера используется convert.

Файл, по которому пользователь щелкнул правой кнопкой мыши, передается нашему сценарию как первый параметр:

```
picture=$1
```

При написании сценариев действий вы можете использовать следующие переменные окружения, которые устанавливаются средой GNOME.

- ▶ NAUTILUS_SCRIPT_SELECTED_FILE_PATHS — имена всех выбранных файлов (в случае если вы выбрали группу файлов, а затем — созданное вами действие) будут разделены символом новой строки (\n). Данная переменная работает только с локальными файлами. Если вы выберете файлы, например, в каталоге FTP-сервера, то эта переменная устанавливаться не будет.
- ▶ NAUTILUS_SCRIPT_SELECTED_URI — эта переменная как раз предназначена для работы с удаленными файлами и содержит URI файлов, разделенных символом новой строки.
- ▶ NAUTILUS_SCRIPT_WINDOW_GEOMETRY — данная переменная содержит геометрию (позицию и размер) текущего окна Nautilus.

KDE

Программа kstart

Программа kstart позволяет управлять запуском приложения в графической среде KDE. Рассмотрим несколько примеров ее использования (подробное описание этой программы вы найдете в справочном руководстве).

Для того чтобы запустить приложение Konqueror на втором виртуальном рабочем столе, достаточно выполнить следующую команду:

```
kstart --desktop 2 konqueror
```

Программу также можно запустить сразу на всех рабочих столах, для этого используется опция alldesktops:

```
kstart --alldesktops konqueror
```

Чтобы запустить все тот же Konqueror и развернуть его во весь экран, используется опция fullscreen:

```
kstart --fullscreen konqueror
```

И, наконец, для запуска на всех рабочих столах приложения noatun в режиме OnTop (всегда поверх всех окон) используется такая команда:

```
kstart --alldesktops --ontop noatun
```

GNOME

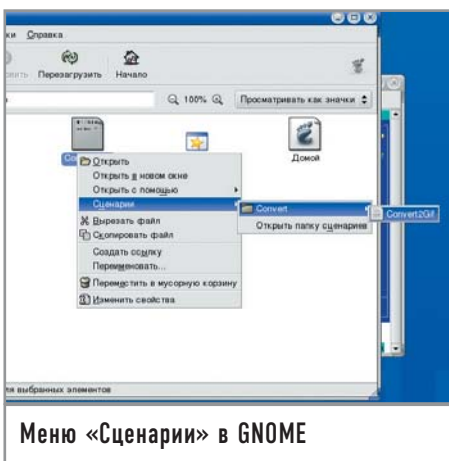
Автоматическая смена обоев

Наверное, многие из нас видели программу Webshots Desktop. Жаль, что она работает только под Windows. Хотелось бы, чтобы у нее была Linux-версия. Но пока ее нет, мы попробуем написать ее аналог.

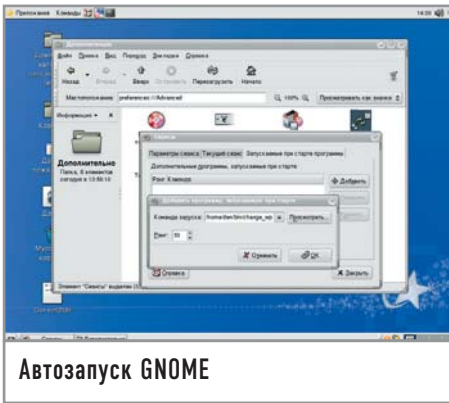
Для тех, кто не знает, что это за программа, вкратце расскажем, как она работает. Существует каталог с обоями рабочего стола — /home/den/Wallpapers/. Каждый раз при запуске GNOME менять обои вручную неудобно. Нужно автоматизировать эту рутинную процедуру, и в этом нам поможет следующий сценарий:

```
#!/bin/bash
export DIR="/home/den/Wallpapers/"
export NUMBER=$RANDOM
export TOTAL=0
for f in `ls $DIR`
do
    let "TOTAL += 1"
done
let "NUMBER %= TOTAL"
export CURRENT=0
for f in `ls $DIR`
do
    if [ $CURRENT = $NUMBER
    ]
    then
        /usr/bin/gconftool-2 -t string -s /desktop/gnome/background/picture_filename $DIR/$f
        break
    fi
    let "CURRENT += 1"
done
```

Особо вникать в этот сценарий не нужно. Достаточно понять, что он с помощью команды gconftool-2 уста-



Меню «Сценарии» в GNOME



навликает файл обоев, который выбирается случайным образом в первой части сценария из каталога, заданного переменной \$DIR.

Сохраните этот файл под именем change_wp и сделайте его исполнимым:

```
$ chmod +x change_wp
```

Для изменения обоев вам достаточно выполнить следующую команду (обои будут изменены мгновенно):

```
./change_wp
```

А теперь, собственно, займемся автоматизацией, ведь вы же не будете вводить эту команду каждый раз при запуске GNOME. Тут можно поступить двумя способами: или поместить вызов этого сценария в таблицу crontab (должен быть запущен демон crond), или добавить команду вызова этого сценария в автозапуск непосредственно GNOME. Crond предлагает больше возможностей. Например, вы можете установить смену обоев каждый час. А в другом случае обои будут меняться только при перезагрузке, то есть всякий раз, когда вы регистрируетесь в GNOME.

Графика

Снимок экрана из командной строки

Снимок экрана (или какого-то окна) можно сделать многими программами: например, очень хороша KSnapshot, входящая в состав KDE. Также снимок экрана можно сделать и с помощью GIMP. Но иногда эти программы не доступны — например, у вас вообще не установлен KDE, а использовать GIMP для снимка экрана/окна, мягко говоря, нерационально.

Сделать снимок экрана можно очень легко и быстро с помощью утилиты

import, которая является частью пакета ImageMagic (этот пакет должен быть у вас установлен). Рассмотрим следующую команду:

```
$ sleep 5; import -window root screen.png
```

Ее можно ввести или в X-терминале, или в окне запуска программы KDE (для вызова окна нажмите сочетание клавиш «Alt+F2»). Данная строка на самом деле состоит из двух команд — sleep и import. Команда sleep генерирует необходимую задержку в секундах. Думаем, пяти секунд хватит, чтобы вы смогли привести экран в надлежащий вид: запустить или активировать нужное окно, выбрать подходящий пункт меню и т. д.

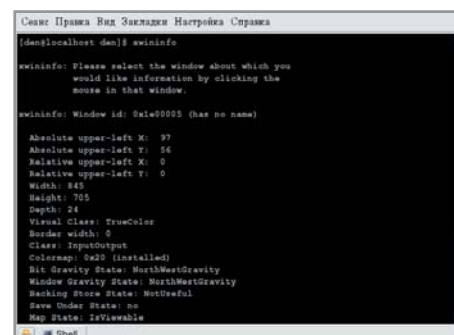
Команда import будет выполнена сразу после обработки sleep, то есть спустя пять секунд. Она делает снимок корневого окна, то есть всего экрана, и записывает его в файл screen.png. Формат PNG наиболее оптимален для создания снимка экрана. Если вам нужен другой формат, например JPG, просто измените расширение результирующего файла:

```
$ sleep 5; import -window root screen.jpg
```

Но не всегда нужен снимок всего экрана, в большинстве случаев необходимо лишь какое-то конкретное окно. Запускать потом приложение GIMP и вырезать нужное окно из общей «фотографии» не очень удобно — проще с самого начала сделать снимок определенного окна.

Первый способ — указать координаты окна, например:

```
$ sleep 5; import -crop 400x300 screen.png
```



Программа XWinInfo. Этот снимок сделан с помощью команды import

Второй способ — указать геометрию окна:

```
$ sleep 5; import -geometry геометрия screen.png
```

Вычислить координаты и геометрию позволяет команда xwininfo.

После запуска программы XWinInfo указатель мыши поменяет свой вид со стрелки на крестик. Вы должны выбрать окно, о котором хотите получить информацию, после чего программа вам их предоставит. Геометрия окна выводится почти в самом конце, поэтому на рисунке она не видна.

И, наконец, рассмотрим третий, самый простой способ сделать снимок определенного окна:

```
$ import window.png
```

Команда sleep не понадобится, поскольку программа import предоставит вам возможность выбрать нужное окно: указатель мыши изменится на крестик, как и в случае с программой import. Щелкните по нужному окну, и буквально через полсекунды в файл window.png будет записан его образ.

Программа import довольно гибкая, рекомендуем прочитать ее man — вы найдете там много интересного.

А что если вам нужно сделать снимок текстовой программы, а не графического окна? Проще всего запустить ее в X-терминале, а затем снимать окно X-терминала, как показано выше. Если обрамление окна X-терминала вам не нужно, его можно вырезать с помощью GIMP.

Администрирование

Клонирование системы

Клонирование — это очень полезный процесс. Предположим, вы только что установили Linux на один компьютер в интернет-зале. Потом посмотрели вокруг и осознали, что вам предстоит повторить этот процесс минимум еще на пяти-семи машинах. Вот тут как раз и целесообразно использовать клонирование — создание точной (побитной) копии исходного носителя. В этом случае в качестве носителя будет выступать корневая файловая система Linux. Клонированная копия называется образом. Мы попробуем сделать точную копию, а потом развернуть ее

на других компьютерах в зале. Надеемся, вы понимаете, что такая операция пройдет корректно, если у всех компьютеров будет одинаковая конфигурация. В случае с интернет-залом обычно так оно и есть. Конечно, можно развернуть образ и на компьютере, конфигурация которого отличается от исходного, но потом он может потребовать дополнительной настройки — вплоть до перекомпиляции ядра. А на все это уйдет намного больше времени, чем на установку системы с нуля, которая на современных компьютерах занимает не более получаса.

Не подумайте, что клонирование полезно только тогда, когда у вас много одинаковых компьютеров. Его целесообразно использовать, даже если у вас всего один компьютер. Например, вы можете сделать образ домашней системы. Если что-то вдруг слетит, вы легко и быстро восстановите исходное состояние системы простым развертыванием образа. Или если у вас что-то случилось с сервером, можно очень быстро «понять» его — ведь на развертывание образа нужно гораздо меньше времени, чем на установку и настройку системы. Время простоя в этом случае будет минимальным.

Итак, приступим непосредственно к клонированию. Перезагрузитесь в однопользовательском режиме. Для этого нужно использовать параметр `single` ядра Linux. Введите команду `mount`, чтобы узнать, какой раздел содержит корневую файловую систему:

```
/dev/hda1 on / type ext3 (rw,noatime)
none on /proc type proc (rw)
none on /sys type sysfs (rw)
```

Корневая файловая система расположена в разделе `/dev/hda1`. Создадим каталог `/mnt/image` — он нам скоро понадобится:

```
# mkdir /mnt/image
```

К этому каталогу нужно подмонтировать носитель, на который будет записываться образ. Понятно, что в директорию `/dev/hda1` записать образ вы не сможете, поскольку образ именно этого раздела и собираетесь сейчас делать. Носителем может быть другой жесткий диск. Мы же будем использовать внешний USB-винчестер. Для его подключения нужно загрузить модуль `usb_stor-`

`age`. Практически во всех новых диспетрибутивах он уже откомпилирован.

```
# modprobe usb_storage
# mount /dev/sda1 /mnt/image
```

Первая команда загружает модуль (если он еще не загружен), а вторая монтирует устройство `/dev/sda1` (это и есть внешний винчестер) к каталогу `/mnt/image`. Все, что осталось сделать, — это смонтировать корневую файловую систему в режиме «Только чтение» и создать образ:

```
# sync
# mount -o remount,ro /
```

Создаем образ `/mnt/image/image.bin` раздела `/dev/hda1`:

```
# dd if=/dev/hda1
of=/mnt/image/image.bin
```

Подробно описывать утилиту `dd` мы не станем — это очень хорошо сделано в справочной системе. Скажем только, что кроме `dd` есть еще и утилита `dd_rescue`, которая при клонировании пропускает плохие секторы.

Теперь рассмотрим, каким образом можно восстановить систему. Для этого вам понадобится загрузочный диск Linux. Подойдет первый диск любого дистрибутива (при условии, что он загрузочный) — Mandrake, Red Hat; можно также использовать Knoppix CD или Gentoo LiveCD.

Если вы используете не LiveCD, а простой загрузочный диск, для перехода в консоль нажмите «Ctrl+Alt+F2». Теперь подмонтируем внешний винчестер:

```
# mkdir /image
# modprobe usb_storage
# mount /dev/sda1 /image
```

Выше приведенные команды (при условии, что второй жесткий диск подключен как Primary Slave) будут выглядеть так:

```
# mkdir /image
# mount /dev/hdb1 /image
```

Теперь вам нужно создать разделы на новом винчестере. Это можно сделать с помощью утилиты `fdisk` или программы установки — как вам удобнее. Если же вы восстанавливаете систему после сбоя, ничего создавать не нужно — все уже есть. Не забудьте только создать раздел подкачки (тип раздела `/dev/hda2` должен быть Linux swap):

```
# mkswap /dev/hda2
```

Далее развернем образ (обратите внимание на параметры `if` и `of` программы

`dd` — теперь их аргументы поменялись местами):

```
# dd if=/foo/image.bin of=/dev/hda1
```

Сейчас нам нужно изменить корневую файловую систему, чтобы попасть «внутри» развернутого образа:

```
# mkdir /install
# mount /dev/hda1 /install
# chroot /install /bin/bash
```

Первая команда создает каталог `install`, вторая монтирует «подопытный» раздел к этому каталогу, а третья изменяет корневую файловую систему. Теперь корнем стал каталог `install`, а в качестве командной оболочки используется директория `/bin/bash`. Все, что нам осталось сделать, — это перезаписать загрузчик. Если вы используете LILO, введите следующую команду:

```
# lilo
```

А если GRUB — то такую:

```
# grub-install /dev/hda
```

Теперь перезагрузите компьютер (команда `reboot`) — ваша система успешно клонирована/восстановлена!

Администрирование

Простое резервное копирование по сети

Предположим, что вы — сетевой администратор. В ваши обязанности входит также и резервное копирование пользовательских данных, то есть каталогов `/home` с каждого компьютера сети. Подходить к каждому компьютеру — не хочется, поэтому лучше автоматизировать процесс. Скопировать каталог `/home/den`, расположенный на компьютере `denis`, можно с помощью этой команды:

```
scp -r backup-den denis:/home/den
```

Параметр `-r` означает, что будут копироваться также и подкаталоги удаленного каталога. Часть команды `backup-den` — это имя каталога, куда будет записана резервная копия; `Denis` — имя компьютера (можно использовать IP-адрес, например, `192.168.1.5`); `/home/den` — это удаленный каталог.

Что же за команда `scp`? Она расшифровывается `secure copy`. Для ее работы нужно, чтобы на компьютере `Denis` был запущен демон `sshd`. В свою очередь, `sshd` спросит вас о пароле и имени пользователя — и вы должны ответить на этот вопрос.

Тюнинг

Повышаем производительность системы

Для повышения производительности системы вам не понадобится перекомпилировать ядро. Данный совет рассчитан только на версию ядра 2.6.

Оптимизацию производительности начнем с установки оптимальных для нас параметров виртуальной памяти.

В псевдофайле `/proc/sys/vm/swappiness` содержится коэффициент подкачки. Что это такое? Предположим, что вы работаете с несколькими (или даже одним) довольно громоздкими приложениями и нечасто переключаетесь между ними. Возможно, вы дизайнер — с утра вы запускаете GIMP и не выходите из него до самого вечера. Также утром вы запускаете xmms — как работать без музыки? Изредка, скажем раз пять-семь за день, вам нужно переключаться между GIMP и xmms, чтобы выбрать новый каталог или новую песню.

Если вы установите большое значение коэффициента подкачки, например 90 или даже 100 (максимальное), то переключение между этими приложениями будет происходить довольно медленно, но зато производительность основного приложения (то есть GIMP) заметно улучшится.

Если вы целый день работаете с небольшими программками и часто переключаетесь между ними, вам лучше установить коэффициент в районе 20 или 30.

Поэкспериментируйте с различными параметрами и выберите оптимальный для себя. Вывести значение файла `/proc/sys/vm/swappiness` можно с помощью команды:

```
# cat /proc/sys/vm/swappiness
```

Значение по умолчанию — 70. Возможно, вам больше подойдет именно такое.

Установить значение (в данном случае 20) можно с помощью команды:

```
# echo "20" > /proc/sys/vm/swappiness
```

Все действия должны осуществляться только с правами суперпользователя (root). Теперь займемся повышением производительности сети. Если у вас обычный домашний компьютер, подключенный к Интернету по *DSL или Ethernet, имеет смысл попробовать выключить некоторые параметры, это можно сделать следующими командами:

```
# echo "0" > /proc/sys/net/ipv4/tcp_sack
```

```
# echo "0" >
```

```
/proc/sys/net/ipv4/tcp_timestamps
```

Если результат вас не устроит, значение параметров можно вернуть («1»).

А теперь приступим к самому важному. У вас есть отличный шанс существенно повысить производительность системы.

Каждой программе, работающей под Linux, время от времени необходим доступ к диску. Ядро Linux определяет, когда именно программа получит доступ к диску. Часть ядра, отвечающая за это, называется планировщиком ввода/вывода. Есть четыре алгоритма его работы.

► Режим по умолчанию (noop) — вряд ли он подойдет для обычного пользователя, несмотря на то, что используется по умолчанию. Рассматривать его мы не будем.

► Упреждающее планирование (Anti-cipatory Scheduling) — при чтении программой данных с диска ядро пытается предугадать, какие данные программа будет читать при следующей операции. Кроме всего прочего эффективность этого алгоритма сильно зависит от логики программы. Параметр ядра — `elevator = as`.

► «Справедливая» очередь (Complete Fairness Queuing) — равные права для всех программ. Ядро равномерно планирует операции ввода/вывода для каждого приложения, здесь нет каких-либо программ, которые могут монополизировать доступ к диску. Если несколько программ одновременно запросят доступ к диску, все они получат ответ. Данный метод во многих случаях позволяет повысить производительность системы, а в некоторых, наоборот, снижает общую производительность — все зависит от конкретной ситуации. Параметр ядра — `elevator = cfq`.

► Deadline-планирование, или планирование крайних сроков (Deadline Queuing) — все приложения, запросившие доступ к диску, ставятся в очередь. Из очереди извлекается одна программа, которая и получает практически монопольный доступ к диску. Пока эта программа работает, все остальные ожидают очереди. По истечении определенного времени планировщик переводит эту программу в состояние ожидания и переключается на другую — следующую по очереди. Теперь вторая программа получает безраздельный доступ к диску. Потом третья, четвертая и т. д. Данный ме-

тод оптимален для сервера баз данных, но не для пользовательского компьютера. Параметр ядра — `elevator = deadline`.

У каждого алгоритма есть преимущества и недостатки. Но только два алгоритма подходят для обычного домашнего компьютера (рабочей станции) — второй и третий. В Интернете вы можете найти данные о том, что для настольного компьютера более подходит второй алгоритм. Решать вам — все познается в сравнении, тем более, что вы ничем не рискуете. Для применения нового алгоритма достаточно указать определенный параметр ядра и немного поработать с системой — если не понравится, всегда можно попробовать другой вариант. Вам не нужно перекомпилировать ядро — необходимо просто пару раз перезагрузить компьютер. Если вы подберете нужный алгоритм, тогда вам понадобится перезаписать загрузчик (в случае с LILO).

Для выбора нужного режима перезагрузите компьютер и при запуске ядра Linux передайте ему один из указанных выше параметров. Например, для выбора упреждающего планирования нужно передать ядру параметр `elevator = as`:

```
linux elevator=as
```

Предположим, что вам больше всего понравился именно этот параметр. Теперь его нужно «утвердить» в файле конфигурации загрузчика. Если вы используете LILO, откройте файл `/etc/lilo.conf` и найдите в нем строку:

```
append = "<параметры ядра>"
```

Добавьте в эту строку параметр `linux elevator = as`:

```
append = "elevator = as (и другие параметры)"
```

Например, файл может выглядеть так:

```
image = /boot/vmlinuz-2.6.9
```

```
label = Linux
```

```
root = /dev/hda1
```

```
append = "elevator = as video =
```

```
vesafb:ywrap,mtrr,1024x768-16@75"
```

Сохраните файл и перезапишите загрузчик командой `lilo`.

Если у вас загрузчик GRUB, перезапись не обязательна. Нужно только отредактировать файл `/boot/grub/grub.conf`:

```
title My Default Linux
```

```
root (hd1,0)
```

```
kernel /boot/vmlinuz-2.6.9 ro root =
```

```
/dev/hda1 elevator = as
```

```
...
```


С чего начиналось ваше знакомство с командной оболочкой? Команды, опции, аргументы... Сколько это продолжалось? Ровно до тех пор, пока вы не услышали о возможности избавиться хотя бы от части этой рутины с помощью скриптов. Как водится, при любом начинании необходима консультация по работе со сценариями командной оболочки. Обратитесь к Advanced Bash Scripting Guide. Это объемное руководство соответствует

печатному объему в 646 листов, предоставляет исчерпывающие сведения о скриптах в bash, содержит более 300 щедро прокомментированных примеров их использования и множество таблиц. Кроме того, вы получите и настоящее введение в базовую технику программирования.

Язык интерфейса: английский

Advanced Bash Scripting Guide

Лицензия: GPL

*Домашняя страница:
www.tldp.org/LDP/abs*



iPod-Linux Installer

Обладатели плееров iPod теперь могут установить Linux на любимое устройство. Для этого необходимо подключить iPod к Macintosh через FireWire, свериться со списком поддерживаемых устройств на странице iPod Linux Project, скачать iPod-Linux Installer, примонтировать устройство, выбрать «iTunes → Preferences → iPod», включить использование диска и запустить программу установки. После перезагрузки iPod при появлении логотипа Apple стоит лишь удерживать кнопку «Назад», и снова появится приветствие знакомого улыбающегося Тух. Вместе с прошивкой вы получите целый ряд новых возможностей: запись аудио

в моно с частотой до 96 кГц, просмотр графических файлов (включая JPEG, GIF и BMP) в двухбитном режиме, игры Othello, Pong, Tetris, Asteroids и Minesweeper, калькулятор, календарь... Беспокойтесь за оригинальную прошивку? А она осталась. Выбираете «Меню → Проигрывание/пауза» и попадаете в привычный интерфейс iPod. Естественно, не стоит рассматривать эту заметку как руководство к незамедлительному действию, помните — осторожность никогда не повредит.

Язык интерфейса: английский

Лицензия: GPL

*Домашняя страница:
www.ipodlinux.org*

gno3dtet

Gnome 3D Tetris — оригинальное решение хорошо известного тетриса, где надо укладывать падающие фигуры в трехмерном пространстве. Здесь можно настроить абсолютно все — от глубины, ширины и длины поля до звукового сопровождения событий игры. Кроме того, есть возможность выбора степени сложности по типам фигур. При потере окном приложения фокуса от простого перемещения

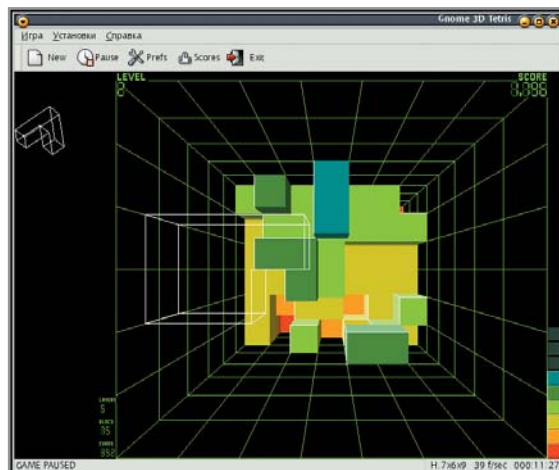
мыши за пределы окна автоматически устанавливается пауза. Игра не рассчитана на поддержку аппаратного ускорения и написана с помощью Gtk+, поддержка OpenGL пока только в планах, однако это не мешает ей уже сегодня выглядеть вполне достойно. Попробуйте — оторваться от этого мощного логического симулятора порой очень трудно.

Язык интерфейса: английский

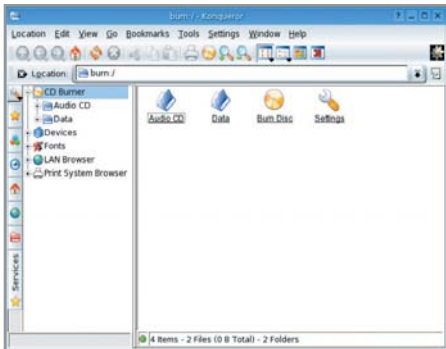
Лицензия: GPL

Домашняя страница:

http://gno3dtet.eseb.net



Запустите Konqueror, введите в строке навигации burn:/ и просто скопируйте



нужные данные в директорию /data (музыкальные композиции — в Audio CD), нажмите «Burn Disc». Вот и все, что от вас требуется. Программа KIO_burn не была задумана специально для Konqueror и фактически совместима с любым приложением KDE, способным работать с системой ioslave. Konqueror — самый простой для этого путь, отсюда и особые отношения KIO_burn с этим файловым менеджером. Для успешной установки и работы программы необходимо: конечно, KDE — автор рекомендует версии начи-

KIO_burn

ная с 3.3.2, aKode из пакета kdemultimedia, естественно, понадобятся cdcrecord и mkisofs, а также Taglib, libaudiofile и DVD+RW-tools. Возможно использование библиотек k3b для определения устройств записи CD и DVD.

Язык интерфейса: русский

Лицензия: GPL

Домашняя страница:

www-users.york.ac.uk/~jrht100/burn

Новый интегрированный пакет KOffice 1.4 воплощает самые современные технологические решения, позволяющие эффективно справляться с различными офисными задачами. В первую очередь стоит отметить, что теперь пользователь имеет пакет приложений, поддерживающих новый открытый офисный формат OpenDocument. Вопреки самым смелым утверждениям, команда разработчиков решила не торопиться с полным переходом на этот формат — пока фильтрам не хватает уверенности в работе и функциональности, «родным» остается собственный формат KOffice, а полный переход на новый формат состоится уже в следующем релизе. Состав KOffice 1.4 претерпел некоторые изменения по программному содержанию — в него включены новые приложения Kexi и Krita.

Kexi — интегрированное окружение для создания и управления базами данных. Разработка и включение такого приложения в состав KOffice мотивированы отсутствием средств быстрой разработки приложений, Rapid Application Development (RAD) для баз данных, достаточно мощных, недорогих, ведомых открытыми стандартами и переносимых на различные программные и аппаратные платформы. На сегодня Kexi полноценно поддерживает SQLite, PostgreSQL и MySQL. В будущем наладится поддержка и других систем, включая интерфейс ODBC. Kexi может рассматриваться как

долгожданный OpenSource-конкурент MS Access, Filemaker и Oracle Forms. Интересно, что версия Kexi, включенная в состав KOffice, по функциональности несколько уступает той, которая доступна в отдельном исполнении. Графический редактор Krita имеет широкие возможности — от простого редактирования фото до создания оригинальной графики. Это первый публичный релиз, с которого начинается долгая дорога нового приложения для художников — по крайней мере, так обозначили перспективу развития Krita разработчики, и конкуренция The GIMP/Photoshop их нисколько не смущает. Уже сейчас Krita имеет собственный набор инструментов, расширяемую гибкую систему управления цветами, основанную на lcms, читает большое число известных графических форматов, включая файлы GIMP и Photoshop, позволяет использовать палитры, кисти, шаблоны, градиенты GIMP и многое другое.

Улучшение нового офисного пакета можно выявить не только по включенным в его состав отдельным приложениям, положительные изменения коснулись продукта в целом. Полностью переписан фильтр Excel — как следствие, импорт файлов Microsoft Excel выглядит очень достойно. Кроме того, в KSpread теперь есть такой модуль

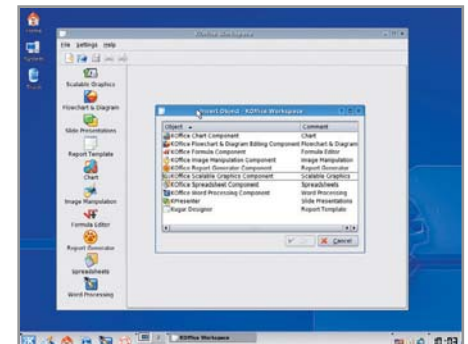
KOffice1.4

как календарь. KChart оснащен новым, существенно усовершенствованным редактором данных, линейной/логарифмической шкалой, имеет поддержку печати и осуществляет экспорт в PNG и SVG. KPresenter может отныне обрабатывать не только слайды, но и мастер-листы. В KWord пересмотрен фильтр импорта документов Word Perfect, а модуль проверки орфографии стал значительно быстрее. Также доступно автозавершение вводимого текста, исполненное в виде всплывающей подсказки. В Kivio улучшена поддержка шаблонов. Не забыли и о «косметике» — например, KOffice Workspace имеет новую панель значков, кнопку «Закрыть», как в Konqueror, а кроме того, поддерживает вкладки.

Язык интерфейса: русский

Лицензия: GPL

Домашняя страница: www.koffice.org





Linux-сервер своими руками

Основное внимание в книге уделяется настройке сервера на совместимых с Red Hat дистрибутивах. Однако цель описать какой-то конкретный дистрибутив изначально не ставилась. Ведь книга называется «Linux-сервер», а не «Сервер Red Hat». Почему именно Red Hat? Потому что большинство популярных на наших просторах дистрибутивов все-таки в большей или меньшей степени являются производными от Red Hat и полностью или частично с ним совместимы. С другой стороны, если описать в книге только Red Hat, она окажется бесполезной для пользователей, у которых установлен другой дистрибутив, например тот же ALT Linux или Mandrake.

Первая часть этого издания представляет собой введение в Linux. Первая глава — это своеобразный курс молодого бойца. В ней вы найдете информацию о сетях, протоколе TCP/IP, IP-адресах, масках и всем том, что просто обязан знать каждый администратор.

Вторая глава посвящена установке Linux. В ней рассматриваются три основных дистрибутива, которые взяты за основу при написании этой книги: Red Hat 9, Fedora Core 3 и Mandrake 10.

Кроме описания установки самой Linux во второй главе также рассказывается о работе с пакетами в Linux.

Вторая часть издания повествует о возможностях операционной системы: рассматриваются файловая система, работа с учетными записями, а также настройка сети — от локальной до DSL-соединения. Причем в последней главе автор пытается объять необъятное: здесь представлены не только последние версии дистрибутивов, но и описаны отличия настройки сети на переходных и самых старых версиях, где вообще нет никаких конфигураторов и все приходится делать вручную.

Третья часть книги посвящена настройке основных программ-серверов: пакет Samba, DHCP-сервер, программы учета и анализа трафика (MRTG и LAN Billing), SSH-сервер, DNS-сервер (BIND), FTP-сервер (wuftp и ProFTP), веб-сервер Apache, SMTP/POP-сервер (Sendmail), сервер новостей INN, прокси-сервер SQUID и SOCKS5. Отдельное внимание

автор уделяет вопросам маршрутизации и защиты сервера.

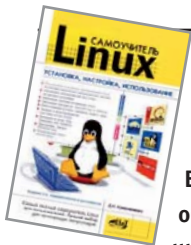
В четвертой части представлены сетевые проекты: сервер времени, кластер на базе PVM, а также описано создание виртуальной частной сети (VPN) под Linux.

Информационная безопасность — это отдельный вопрос, которому посвящена вся пятая часть книги. Здесь повествуется о IPChains, IPTables, системе LIDS, антивирусной защите сервера и сети, работе с RAID-массивами, создании резервных копий и дерева chroot.

В оставшихся двух частях вы найдете практические примеры по настройке серверов, а также описание синтаксиса командной оболочки bash. Без знания bash не обойдется ни один администратор. В этих же частях подробно описана компиляция ядра.

Таким образом, в представленной книге найдутся материалы, интересные как начинающим пользователям, так и опытным администраторам.

Колисниченко Д. Н. Linux-сервер своими руками. — 3-е изд. — СПб., 2005



Самоучитель Linux

Еще одна книга Дениса Колисниченко попала в наш обзор не случайно. Если рассмотренное выше издание больше ориентировано на создание и поддержку

Linux на сервере, то этот самоучитель посвящен домашнему использованию этой операционной системы.

Как обычно, в самом начале описывается установка системы.

В отличие от «Linux-сервера» в «Самоучителе Linux» описывается установка других дистрибутивов — Red Hat Linux, ASP Linux и Fedora Core. Последний дистрибутив описан довольно подробно, в том числе уделено внимание решению различных проблем, которые то и дело возникают при его установке, а также описан процесс обновления Red Hat 7/8/9 до FC. Далее от инструкций по установке системы автор переходит непосредственно к описанию особенностей работы с файловой системой, KDE, GNOME.

Вторая часть книги посвящена работе с офисными пакетами OpenOffice и KOffice. Причем описание не сводится просто к традиционному сравнению типа «это — аналог Word, а это — аналог Excel». Ознакомившись с материалами этой части, вы действительно научитесь работать с OpenOffice и KOffice. Кроме этого рассматриваются и другие офисные программы — электронные

таблицы Gnumeric, издательская система Scribus и прочее.

В следующей части описана работа с мультимедиаданными — от создания картинок до прожига CD. Очень подробно рассказано о работе с программами GIMP и MainActor. Первая, как вы знаете, является отличным графическим редактором, вполне заменяющим Photoshop в домашних условиях. А MainActor подходит для любительского видеомонтажа в Linux. Кроме всего прочего в этой части описана настройка ТВ-тюнера под управлением Linux. Особое внимание в книге уделено настройке сети и подключению к Интернету. Ведь что нужно домашнему пользователю? По большому счету, три вещи, не считая игрушек: средство для набора и форматирования текстов, программа для воспроизведения музыки и видео и, конечно же, Интернет. В данной книге вы обязательно найдете все, что вам нужно для комфортной работы, в том числе и информацию по настройке сети на разных дистрибутивах, таких как ALT Linux, а также различные способы подключения к Интернету — через сеть, модем или DSL-соединение. В заключение можно сказать, что данная книга идеально подойдет начинающему пользователю и станет хорошим помощником в освоении Linux.

Колисниченко Д. Н. Самоучитель Linux. Установка, настройка, использование. — 3-е изд., переработанное и дополненное. — СПб., 2004. — 656 с.

Это скорее идеологическая и занимательная книга, чем учебник. Одно название чего стоит: шутка ли, рассказывать обо всех Unix-подобных системах скопом. В книге много внимания уделяется придуманной автором концепции различения «проективных» (основанных на предварительной настройке человеком проектах будущей системы) и «процедурных» (основанных на выполнении человеком требуемых процедур в заранее заданной системе) методов. Идея, что Unix — система «проективная», проходит красной нитью через все разделы книги. Автор рассматривает сложные и проблемные аспекты Unix-подобных систем и показывает, как их можно решить с помощью «проективного» подхода. Итак, читать данную книгу можно даже не имея под рукой компьютера: литературный стиль и жизненные примеры способствуют хорошему усвоению. Другой вариант — читать по чуть-чуть, раз в неделю, посвящая остальное время практической работе с Unix-системой и изучению документации.

Цитата из LJ Даниила Борисовича Гинсбурга (www.livejournal.com/users/dbg/20539.html):

«Автор очень старательно и аккуратно обходит любой намек на возможное наличие в системе нелогичностей и странностей, не



Операционная система Unix

говоря уже о коварных ловушках. Даже когда повествование само идет к тому, чтобы упомянуть подводные грабли, автор останавливается, откашливается и продолжает уже с другого места. Даже команда [, в описании автора (стр. 168) выглядит как образец логичности и беспроблемности (но мы-то знаем, что худшее уродство надо еще поискать). Итого, в книге описывается какая-то странная система, отличающаяся стройностью, логичностью и непротиворечивостью, внимательно спроектированная и аккуратно реализованная. Словом, описан какой-то анти-юникс. Реальный же юникс спроектирован не был — он мутировал из запускатки Space Travel до современного своего состояния в очень жестких и недружелюбных условиях, в которых речь шла не о красоте и идеологической стройности, а банальном выживании. Соответственно, получился покрытый шрамами и редкой шерстью пятилапое существо с двумя сердцами и тремя желудками, очень живучее, но совершенно безалаберное. За это мы его и любим».

Курячий Г.В. Операционная система UNIX. Курс лекций. — М., 2004

Книга «Операционная система Linux» написана в жанре «Linux для самых маленьких». Слово «маленькие» здесь не стоит понимать как «идиоты», «домохозяйки» или «лилипуть». Авторы, напротив, считают, что «ребенок — это мало информированный взрослый», поэтому эта книга так похожа на учебник, не хватает только задач и упражнений. Впрочем, внутри самого текста имеется изрядное число примеров, которые, кстати, даются от лица двух вымышленных персонажей — Мефодия Кашина (новичка) и Льва Гуревича (опытного линуксоида). Предполагается, что за время работы с учебником (курс рассчитан на полгода, а если «расходовать экономно», то и на год) Мефодий станет



Операционная система Linux

свободно ориентироваться во вселенной Linux.

Авторы не стали использовать популярный нынче стиль «Синхротрон для чайников» и подошли к изучению Linux, традиционно ориентируясь на Unix. Изложение начинается с описания терминального сеанса, командной строки, работы с процессами и правами доступа. От обработки текстов и настройки собственного окружения пользователя авторы переходят к настройке системы в целом, работе с пакетами, сетью, устройствами графической подсистемы. Последние лекции учебни-

ка обзорные, они описывают современное состояние дел с программным обеспечением (это чуть ли не самая большая лекция в книге), бегло рассказывают о серверных возможностях Linux, истории ее создания и аспектах лицензирования. Книга, конечно, адресована новичкам. Она явно рассчитана на использование в лекционном режиме (два часа в неделю), так как количество информации в каждой главе довольно велико и требует непременно закрепления материала посредством практических занятий.

Курячий Г. В., Маслинский К. А. Операционная система Linux. — М.: 2005

Об авторах

Курячий Георгий Владимирович

В 1992 году закончил факультет вычислительной математики и кибернетики (ВМиК) Московского государственного университета. Работал в компьютерной скорой помощи, преподавал информатику в вечерней математической школе при ВМиК МГУ. В настоящее время — консультант факультета по вопросам системного администрирования, читает спецкурс по Unix-системам, органи-

затор проблемного семинара «UNIX». С 2002 года — сотрудник компании ALT Linux, где занимает попеременно различные должности.

Маслинский Кирилл Александрович

В 2002 году закончил филологический факультет Санкт-Петербургского государственного университета. Работал в области издательского дела. На данный момент занимается подготовкой и выпуском технической документации в компании ALT Linux. Интересы в области информационных техно-

логий: свободное программное обеспечение, языки разметки и структура документа, автоматизация издательского процесса.

Колисниченко Денис Николаевич

Проживает в Кировограде (Украина), учится на кафедре информатики физико-математического факультета Кировоградского государственного педагогического университета. В настоящее время работает программистом и системным администратором на частном предприятии «Ник-МТД».