ХАКЕР
СПЕЦ
08|69|АВГУСТ 2006
ЕЖЕМЕСЯЧНЫЙ
ТЕМАТИЧЕСКИЙ
КОМПЬЮТЕРНЫЙ
ЖУРНАЛ

ЧЕРТОВА ОС

БОЖЕСТВЕННЫЕ ДОСТОИНСТВА xBSD

(game)land hi-fun media
9 771609 102006 08>
publishing for enthusiasts

NT computer

AMD

СДЕЛАЙТЕ РАЗУМНЫЙ ВЫБОР

Ваш новый виртуальный дом ждет Вас!

www.nt.ru

Процессор AMD Athlon™ 64 - передовая производительность для игр, видео и музыки

AMD 64 Athlon™

www.amd.ru

Надежные компьютеры для любых задач.
Модельный ряд на все запросы и возможности. 3 года гарантии.

Компьютеры марки <NT> на базе процессора АМД Athlon ™ 64 спрашивайте в магазинах Федеральной сети компьютерных центров POLARIS.
Оптовые поставки (495) 970 1930. Сеть региональных филиалов.

intro

Вообще, многие вещи можно объяснить одержимостью. Так, например, и поступали в свое время товарищи Яков Шпренгер и Генрих Крамер. Например, господин Яков, хоть и был деканом Кельнского университета, не стеснялся писать в Malleus Maleficarum («Молот Ведьм») всякие хитрые рецепты про то, как бороться с ворожеями, одержимыми, товарищами, вступавшими в плотские сношения с суккубами и инкубами и т.д. В наше время, кстати, тоже встречается немало одержимых людей. Они одержимы программированием, вступают в плотские сношения с операционными системами и жить не могут без своего ноутбука. Почему же мы не жжем их на кострах (хватит уже глупо хихикать при слове «жжОм»), не раздираем пыточными когтями или не скармливаем диким зверям? А это наша национальная традиция. Ну знаешь, толпа раздолбаев и несколько одержимых-супер-профи-админов-программеров-инженеров, на которых весь отдел и держится. Правда, раздолбаев тоже никто зверям не скармливает по неясной причине.
Так к чему я это пишу? Ага, вспомнил. Номер-то непростой, ведь он должен помочь начинающему юниксоиду!

Человеку, который решил изучить Free/OpenBSD, одержимым быть строго запрещается! Разрешено только одно — просто быть IT-профессионалом без закидонов.

Александр Лозовский

СПЕЦ

ЕЖЕМЕСЯЧНЫЙ
ТЕМАТИЧЕСКИЙ
КОМПЬЮТЕРНЫЙ
ЖУРНАЛ

08|69|2006

WWW.XAKEP.RU

РЕДАКЦИЯ
Главный редактор
Николай «AvaLANche» Черепанов (avalanche@real.xakep.ru)
Выпускающие редакторы
Александр «Dr.Klouniz» Лозовский (alexander@real.xakep.ru)
Андрей Каролик (andrusha@real.xakep.ru)
Редактор CD/OFFTOPIC
Иван «SkyWriter» Касатенко (sky@real.xakep.ru)
Литературный редактор
Анна Большова (bolshova@gameland.ru)
Арт-директор
Иван Васин (vasin@real.xakep.ru)
Дизайнер
Наталья Жукова (zhukova@real.xakep.ru)
Цветокорректор
Александр Киселев

РЕКЛАМА
Директор по рекламе ИД (game)land
Игорь Пискунов (igor@gameland.ru)
Руководитель отдела рекламы цифровой группы
Ольга Басова (olga@gameland.ru)
Менеджеры отдела
Ольга Емельянцева (olgaeml@gameland.ru)
Евгения Горячева (goryacheva@gameland.ru)
Оксана Алехина (alekhina@gameland.ru)
Менеджер по работе с сетевыми РА, корпоративные продажи
Максим Григорьев (grigoriev@gameland.ru)
Трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

РАСПРОСТРАНЕНИЕ
Директор отдела дистрибуции и маркетинга
Владимир Смирнов (vladimir@gameland.ru)
Оптовое распространение
Андрей Степанов (andrey@gameland.ru)
Подписка
Алексей Попов (popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

PUBLISHING
Издатель
Борис Скворцов (boris@gameland.ru)
Редакционный директор
Александр Сидоровский (sidorovsky@gameland.ru)
Учредитель
ООО «Гейм Лэнд»
Директор
Дмитрий Агарунов (dmitri@gameland.ru)
Финансовый директор
Елена Дианова (dianova@gameland.ru)

ГОРЯЧАЯ ЛИНИЯ ПО ПОДПИСКЕ
тел.: 8 (800) 200.3.999 (бесплатно для звонящих из России)
ДЛЯ ПИСЕМ
101000, Москва, Главпочтамт, а/я 652, Хакер Спец
spec@real.xakep.ru
http://www.xakep.ru

Отпечатано в типографии «ScanWeb», Финляндия
Зарегистрировано в Министерстве Российской Федерации
по делам печати, телерадиовещанию
и средствам массовых коммуникаций
ПИ № 77-12014 от 4 марта 2002 г.
Тираж 42 000 экземпляров.
Цена договорная.

ЧИСТИЛИЩЕ

АДСКАЯ СМЕСЬ

SPECIAL DELIVERY

АНДРЕЙ МАТВЕЕВ

ЭКСПЕРТ НОМЕРА КЕРНЕЛ-ХАКЕР,
ЮНИКС-ГУРУ, СПЕЦИАЛИСТ
ПО СТРОИТЕЛЬСТВУ И МЕЛИОРАЦИИ,
А ТАКЖЕ КРАСНОЗНАМЕННЫЙ РЕДАКТОР
БРАТСКОГО ЖУРНАЛА «ХАКЕР»

offtopic

HARD

SOFT

CREW

STORY

cd!

ЗДРАВСТВУЙ, МАЛЫШ! ДАВНО ЛИ ТЫ УСТАНАВЛИВАЛ LINUX? ИЛИ ПЕРЕУСТАНАВЛИВАЛ WINDOWS? НЕДАВНО? ЧТО Ж, КОМПЬЮТЕРНОЕ СООБЩЕСТВО СОЧИНИЛО ДЛЯ ТЕБЯ ЕЩЕ ОДНУ СКАЗКУ ПОД НАЗВАНИЕМ BSD. У НЕЕ ЕСТЬ МНОГО БРАТИШЕК И СЕСТРЕНОК: FREEBSD, OPENBSD, NETBSD, ДА-МАЛО-ЛИ-ЕЩЕ-ЧТО-BSD. ВСЕ ОНИ ПОХОЖИ И ВСЕ ПО-СВОЕМУ РАЗНЫЕ. ПУСТЬ ЖЕ СДЕЛАННЫЙ С ЛЮБОВЬЮ СПЕЦ С ОЧЕРЕДНЫМ ДИСКОМ СТАНЕТ ТВОИМ ПУТЕВОДИТЕЛЕМ В ЗАМЕЧАТЕЛЬНЫЙ МИР BSD-СИСТЕМ!

FREEBSD

- Live! дистрибутив Frenzy 1.0
- m0n0bsd 4.9
- m0n0wall 1.22

OPENBSD

- Дистрибутив OpenBSD 3.9

ТОЖЕ BSD

- Biew 5.62
- Apache 2.2.2
- LDE 2.6.1
- OpenSSL 0.9.8b
- SendMail 8.13.7

СОФТ ОТ NONAME

- CDCheck 3.1.12.0
- Google Video Player 1.0.1.0 Beta
- REAPER 0.977
- phpMyAdmin 2.8.2 Final
- WinBackup Pro v2.20
- WinRAR Version 3.60 beta 6
- Photo Collage v1.40
- Catalyst 6.6
- Fresh UI 7.62
- jetAudio 6.2.6.8330 Plus VX
- MemOptimizer 3.01
- WinLock 4.45 Pro
- Opera 9.01 Test

ВСЕГДА ПОБЕЖДАЮТ

PROgram For InTegration

Вниманию дилеров!
Регистрируйтесь прямо сейчас:
www.fdg.fujitsu.com/profit

Новое поколение жестких дисков Fujitsu максимальной производительности

Fujitsu 3,5" Enterprise ставят невиданные рекорды. **Высокоскоростные жесткие диски** новой серии обладают значительно более высокой производительностью. С емкостью до 300 Гб, внутренней скоростью передачи данных до 132 Мбайт/с, скоростью вращения 10 000 и 15 000 об/мин как для SCSI, так и для Fibre Channel, они снова впереди всех!

Fujitsu 2,5" Enterprise с интерфейсом SAS (Serial Attached SCSI) воплотили в себе высшие достижения новейших технологий хранения данных – скорость вращения **10 000 об/мин**, емкость до **73,5 Гб**, надежность до **1,4 млн. часов безотказной работы** – и предназначаются для самых современных 1U серверов, blade-серверов и массивов хранения данных.

Fujitsu 2,5" Mobile с интерфейсами PATA и SATA, скоростью вращения до 5400 об/мин – специально разработаны для ноутбуков и мобильных накопителей и относятся к **самым быстрым, эффективным и компактным** в своем классе. Теперь емкость дисков этой серии **до 160 Гб**, и признанный технологический лидер Fujitsu снова далеко впереди!

ПИРИТ — официальный дистрибутор Fujitsu

ПИРИТ-Дистрибуция (опт):
(495) 97-43210
Компьютерный салон ПИРИТ: (495) 785-55-54
ПИРИТ Санкт-Петербург (опт): (812) 712-65-02

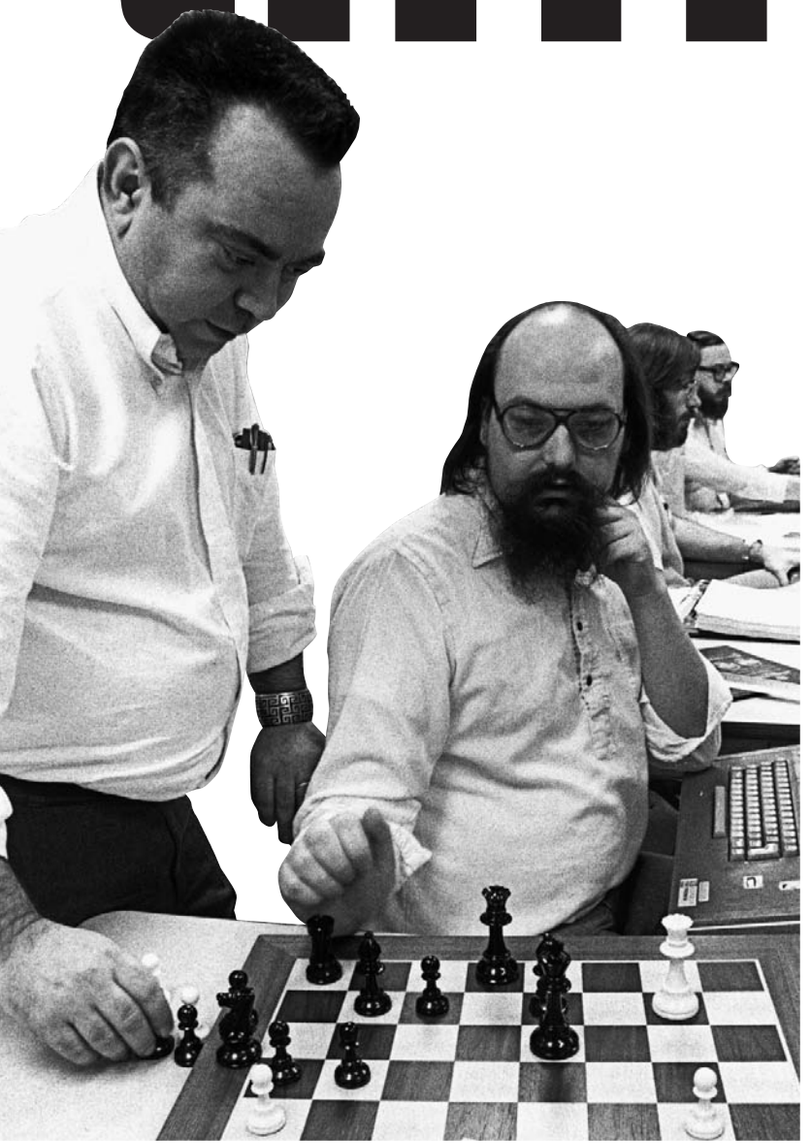
ПИРИТ
www.pirit.ru
www.ddp.ru

FUJITSU
THE POSSIBILITIES ARE INFINITE

OEM-партнеры: InPrice Data Systems – (495)788-1-788, www.inprice.ru · АК-Цент Микросистемс – (495) 232-02-81, www.ak-cent.ru

timeline

АНДРЕЙ КАРОЛИК
{andrusha@real.xakep.ru}



1969

Кен Томпсон написал первую версию новой операционной системы, стремясь реализовать идеи, положенные в основу MULTICS (одна из первых компьютерных операционных систем с разделением времени, MULTIplexed Information and Computing Service). А Брайан Керниган придумал для нее название — UNICS (UNIplexed Information and Computing System). Позже название сократилось до UNIX.

1984

В Массачусетском технологическом институте (MIT) разработана X Window System — оконная система для растровых дисплеев, обеспечивающая стандартные инструменты и протоколы для построения графических интерфейсов. Почти все современные ОС поддерживают X Window System, но в основном она закрепилась в UNIX-подобных системах. X Window System часто называют X11 или просто X, неформально «иксы».

1984

Была впервые представлена Mac OS. Она была совершенством современного программирования. Система изначально разрабатывалась с расчетом на то, чтобы любой неквалифицированный пользователь мог уже через несколько минут начать на ней работать. Другое дело, что технические ограничения, заложенные в систему, не позволяли ей называться «современной». Вплоть до Mac OS 9 она предназначалась для одного пользователя, который бы работал с одним приложением на одном компьютере.

1985

Ричардом Столлмэном основана некоммерческая организация Free Software Foundation (фонд свободного программного обеспечения) для поддержки движения свободного программного обеспечения и проекта GNU. В годы основания средства фонда использовались, в первую очередь, для найма разработчиков для написания свободных программ. Сейчас свободное программное обеспечение создается многими компаниями и частными лицами, поэтому фонд работает в основном над юридическими и организационными вопросами в области свободного ПО.

1991

Финский студент Линус Торвальдс начал разработку ядра операционной системы Linux (Линукс). Большинство кода написано им на C, с некоторыми расширениями GNU C, остальное — на ассемблере, с использованием синтаксиса GNU Assembler «AT&T». Распространяется Линукс свободно на условиях лицензии GNU General Public License. Эту лицензию Линус Торвальдс выбрал практически сразу после того, как стало понятно, что его увлечение — не просто хобби, и Линукс становится популярным во всем мире.

1993

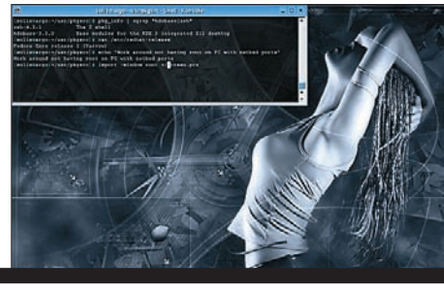
Появилась первая версия NetBSD, которая называлась NetBSD 0.8. Она основывалась на исходном коде системы 4.3BSD Lite, разработанной университетом Berkeley, и системе 386BSD, которая стала первым вариантом BSD Unix, способным работать на процессорах Intel 386. NetBSD впитывала самые лучшие идеи из всех веток BSD-систем. Эти идеи впоследствии трансформировались и совершенствовались энтузиастами, работающими над развитием NetBSD. Лозунгом системы стала фраза «если внутри этой штуки есть процессор, значит, мы будем на нем работать».

1993

Отпочковавшись от неофициальной версии системы 386BSD Patchkit, родился проект FreeBSD. Группа разработчиков состояла из трех координаторов: Нейта Уильямса (Nate Williams), Рода Граймса (Rod Grimes) и Джордана Хаббарда (Jordan Hubbard). Первый дистрибутив базировался на системе 4.3BSD-Lite, созданной в Калифорнийском университете в Беркли, с компонентами от 386BSD и Free Software Foundation.

1995

В результате раскола в команде разработчиков от NetBSD отделился проект OpenBSD. Тео де Раадт (Theo de Raadt, один из четырех основателей NetBSD) был вынужден покинуть проект, так как, по его мнению, команда уделяла недостаточно внимания безопасности системы. OpenBSD отличается от других свободных BSD-систем системой разработки. Никакой код не может попасть в систему извне случайно, любые изменения просматриваются ответственными за соответствующую часть системы людьми. Любая ошибка, найденная в одном месте, вызывает пересмотр всего аналогичного кода. Также OpenBSD уделяет много внимания качеству документации.



1999

Apple провозгласила разработку в рамках «открытых кодов» одним из ключевых принципов своей стратегии, появились исходные тексты первой версии нового ядра (Darwin). Для разработки Apple взяла за основу открытую версию операционной системы UNIX — BSD 4.4. Будучи основанным на BSD UNIX, Darwin поддерживает все возможности BSD: в него включена полная поддержка стандарта POSIX, используется UNIX-модель процессов, поддерживаются потоки UNIX, что позволяет назвать Mac OS X не только многозадачной, но и многопотоковой системой, в которой каждое приложение имеет ряд параллельно выполняемых задач.

2006

Пока открытое и закрытое программное обеспечение не могут стать единым целым. Они остаются двумя разными направлениями, хотя между ними происходят активные интеграционные процессы. Число приверженцев открытого программного обеспечения возрастает как среди пользователей, так и среди разработчиков новых продуктов. Более 90% рынка Linux делят две компании — Red Hat и Novell. В отличие от ситуации с Unix, когда все разработчики работают отдельно, создавая свои версии. В долгосрочной перспективе система Unix может вообще исчезнуть с рынка, и на нем останутся Linux и Windows.

8

ЧИСТИЛИЩЕ

в разделе:

семь кругов xBSD

ИСТОРИЧЕСКИЙ РАКУРС

СЕМЕЙСТВО XBSD С ОГРОМНОЙ СКОРОСТЬЮ ДВИЖЕТСЯ ПО УЗКОЙ КОЛЕЕ СВОЕЙ РЫНОЧНОЙ НИШИ. СЛЕВА — ОТВЕСНАЯ СКАЛА КОММЕРЧЕСКИХ UNIX-СИСТЕМ, СПРАВА — КРУТОЙ ОБРЫВ В LINUX. ЧТОБЫ ВЫБРАТЬ ДИСТРИБУТИВ СВОЕЙ МЕЧТЫ, НЕОБХОДИМО НЕ ТОЛЬКО ИЗУЧИТЬ FEATURE-LIST, НО И РАССМОТРЕТЬ ИСТОРИЧЕСКИЙ АСПЕКТ, ПОСЛЕ ЧЕГО СТАНЕТ ЯСНО, ПОЧЕМУ ДЛЯ FREEBSD ЕСТЬ ДРАЙВЕРА ОТ NVIDIA, А ДЛЯ ОСТАЛЬНЫХ XBSD — НЕТ

КРИС КАСПЕРСКИ АКА МЫЩЪХ

→ **введение.** В 1965 году три компании (Bell Labs, General Electric's, Ford) и Массачусетский технологический институт вплотную занялись дорогостоящими экспериментами, целью которых было создание универсальной, переносимой, многопользовательской, высокопроизводительной операционной системы. Для этого проекта General Electric выделила высокопроизводительную 36-разрядную машину GE-645 с неплохим даже по сегодняшним меркам процессором, оснащенную превосходной канальной подсистемой ввода/вывода (совершенно непозволительной для тех времен роскошью).

В ходе проекта, получившего название MULTICS (Multiplexed Information-n-Computing Service), была реализована система, поддерживающая виртуальную память с сегментно-страничной организацией, с раздельными сегментами данных и кода, имеющих набор атрибутов защиты, определяющих привилегии доступа; динамическое связывание модулей в ходе выполнения программы с механизмом «расщепления» разделяемых страниц при записи (copy-on-write в терминологии NT); иерархическую файловую систему, объединяющую в одну логическую древовидную структуру файлы, физически расположенные на разных носителях и поддерживающую файлы, проецируемые в память; оконную подсистему и ряд других идей, определивших архитектуру ОСей начала XXI века.

МЭТТ ДИЛЛОН

«НЕКОТОРЫЕ СЧИТАЮТ BSD «СТАРОЙ» ОПЕРАЦИОННОЙ СИСТЕМОЙ, НО ТЕ, КТО РАБОТАЕТ НАД НЕЙ, ВИДЯТ ЕЕ СКОРЕЕ СИСТЕМОЙ СО «ЗРЕЛЫМ КОДОМ»

«САМОЙ БОЛЬШОЙ ОШИБКОЙ, КОТОРУЮ МОЖЕТ ДОПУСТИТЬ ПРОГРАММИСТ, ЯВЛЯЕТСЯ ИГНОРИРОВАНИЕ ИСТОРИИ, И ЭТО ИМЕННО ТА ОШИБКА, КОТОРУЮ СДЕЛАЛИ МНОГИЕ РАЗРАБОТЧИКИ СОВРЕМЕННЫХ ОПЕРАЦИОННЫХ СИСТЕМ»

Набитая под завязку передовыми технологиями, система оказалась необычайно прожорливой и для эффективной работы требовала оборудования астрономической стоимости. Фактически единственным пользователем MULTICS оказалась компания Ford, поэтому руководство Bell Labs сочло затею провалившейся и в апреле 1969 приняло решение о выходе из проекта, отозвав своих ведущих разработчиков, среди которых оказались Деннис Ритчи, Кен Томпсон, Мак Илрой и Джон Осанна.

Движимые желанием использовать накопленный опыт для создания дешевого и нетребовательного к аппаратным ресурсам усеченного варианта MULTICS, они обратились к руководству Bell Labs с просьбой приобрести компьютер среднего класса и выделить некоторую сумму под проект. Компания, разочарованная провалом MULTICS, финансировать эту затею наотрез отказалась, но все-таки предоставила Томпсону незадействованный PDP-7, для которого не существовало ни достойного ассемблера, ни библиотек для поддержки вычислений с плавающей точкой. Словом, не было ни хрена, и весь инструментарий пришлось создавать буквально с нуля...

➔ **UNIX.** К началу 1970 года система, названная UNICS (Uniplexed Information & Computing Service) и написанная на чистом ассемблере, уже поддерживала примитивный ввод/вывод и несла на своем борту набор вспомогательных утилит для копирования, удаления, редактирования файлов и примитивного командного интерпретатора. Позже «CS» заменили на «X», и система превратилась в UNIX.

Деннис Ритчи и Кен Томпсон за PDP

Компания высоко оценила работу Томпсона и предложила ему перенести UNIX на PDP-11, что заставило задуматься над повышением мобильности, то есть свести использование ассемблера к минимуму и везде, где только возможно, задействовать высокоуровневые языки, из которых тогда наибольшей популярностью пользовались Фортран, ориентированный на решение вычислительных задач, и Би, привлекающий простотой и легкостью изучения, наглядностью листингов и неплохой производительностью. Так что выбор остановили на нем.

Вторая версия UNIX появилась в 1972 году. Главным нововведением стала поддержка конвейера (pipe), позаимствованная Илроем из операционной системы DTSS (Dartmouth time-sharing System). Использование интерпретируемого языка Би заметно ухудшило производительность системы, поэтому Ритчи и Томпсон решили написать компилятор, попутно ликвидируя наиболее существенные недостатки Би (например, отсутствие типов). Так появился Си, но это уже совсем другая история...

Третья (промежуточная) версия UNIX отличалась хорошей производительностью, практически не уступающей версии, написанной на ассемблере, но при этом для ее создания требовалось значительно меньше усилий, и она не была связана с какой-то одной конкретной архитектурой. Из 13000 строк операционной системы лишь 800 принадлежали низкоуровневым модулям, написанным на ассемблере.

Четвертая версия UNIX, созданная в 1974 году, получила статус «официальной» и, с одобрения руководства, стала применяться внутри компании. Даже по тем временам система представляла довольно убогое зрелище. Виртуальная память не поддерживалась, динамическое связывание отсутствовало, а файловая система при интенсивном использовании за счет фрагментации могла терять до 60% дискового пространства и ограничивала длину имен всего 14 символами, поэтому в основном UNIX использовалась для управления цифровыми АТС и как текстовый процессор.

Системой заинтересовались и другие компании, но антимонопольное законодательство запрещало Bell Labs заниматься каким-либо другим бизнесом, кроме телефонии, поэтому UNIX распространялась без рекламы и сопровождения по чисто символической цене, сопоставимой с ценой носителя.

Первая сторонняя инсталляция UNIX вне Bell Labs была осуществлена Нилом Граундвотером из компании New York Telephone, после чего на Bell Labs обрушился шквал запросов на UNIX. Приблизительно в это же время на открытом симпозиуме ACM прошла первая презентация операционной системы UNIX, сопровождаемая докладами Томпосна, которые произвели неизгладимое впечатление на профессора Берклиевского университета Фабри. Ему удалось убедить собственное руководство в необходимости приобретения PDP 11 и заполучить магнитную ленту с исходными текстами последней на тот момент шестой версией UNIX вместе с лицензией, дающей право на «исследование и доработку».

Так произошло разделение UNIX'а на коммерческую и некоммерческую ветви. Из первой выросли SunOS, HP-UX, AIX, Solaris и т. д. Из второй — FreeBSD, NetBSD, OpenBSD и другие BSD-клоны, о которых мы и будем говорить.

➔ **BSD.** Первая инсталляция UNIX в Беркли была осуществлена в 1974 году на PDP-11, и с этого момента система неуклонно обрастала новым софтом, создаваемым студентами и преподавателями для своих собственных целей. К 1977 году софта накопилось столько, что Билл Джой (в то время аспирант) собрал его в одну кучу и записал на ленту, которую высылал всем желающим под названием 1BSD (Berkeley Software Distribution — распространение софта университета Беркли). Этот «сборник» не был полноценной операционной системой и ставился поверх UNIX 6, которую приходилось приобретать непосредственно у Bell Labs.

Следующая версия 2BSD, выпущенная в 1978 году, становилась поверх UNIX 7 и включала в себя пару новых утилит: текстовый редактор vi и C shell.

В 1978 году в Берлки был установлен первый компьютер семейства VAX, и с этого момента история развития BSD резко изменилась. Официальный UNIX-порт на VAX (UNIX/32V) не использовал всех преимуществ этой системы (прежде всего — страничной организации виртуальной памяти), поэтому студенты практически полностью переписали 32V-ядро и перенесли утилиты из 2BSD, получив к концу 1979 года законченную операционную систему, названную Virtual VAX/UNIX или VMUNIX, она же — 3BSD.

Успех 3BSD привел к тому, что дальнейшая разработка системы финансировалась агентством DARPA, одной из задач которого было создание сети, способной работать даже в условиях ядерной войны (тогда она казалась неизбежной). От Била Джоя BSD отошла к специально сформированной исследовательской группе CSRG (Computer Systems Research Group), выпустившей в 1980 году 4BSD, содержавшую множество мелких улучшений, но не предлагавшую ничего принципиально нового.

Революция свершилась лишь с появлением 4.2 BSD (1983 год), включавшей в себя черновой TCP/IP-стек, улучшенную файловую систему FFS и чертенка по имени Beastie (Бистли) с вилами в руках.

В процессе переноса 4.3 BSD на Power 6/32, завершенном к концу 1988 года, произошла реструктуризация исходного кода с выделением системно-зависимых частей в отдельный слой, что значительно увеличивало мобильность системы. Однако 4.3 BSD по-прежнему базировалась на UNIX, и ее использование требовало лицензии от Bell Labs (точнее от AT&T, к которой после раскола фирмы отошли все права). А между тем коли-

Таблица 1. Сводная информация по xBSD системам

	разработчик	первый релиз	на чем основана	последний резил	цена, $	тип лицензии	назначение
FreeBSD	The FreeBSD Project	декабрь 1993	386BSD, 4.4BSD-Lite	6.1 (8.06.2006)	free	BSD	серверы, рабочие станции, сетевые приложения
OpenBSD	The OpenBSD Project	октябрь 1995	NetBSD 1.0	3.9 (1.06.2006)	free	BSD, see detailed policy	серверы, рабочие станции, сетевые приложения, встраиваемые устройства
NetBSD	The NetBSD Project	май 1993	386BSD, 4.4BSD-Lite	3.0 (23.12.2005)	free	BSD	серверы, сетевые приложения, встраиваемые устройства
386BSD 3	William and Lynne Jolitz	март 1992	4.3BSD Net/2	1.0 (1994)	free	BSD	—
BSD/OS (BSD/386) 3	BSDi, Wind River Systems	март 1993	4.3BSD Net/2, 4.4BSD	5.1 (октябрь 2003)	—	проприетарная	—
SunOS 3	Sun Microsystems	1982	4.xBSD, UNIX System V[20]	4.1.4 (ноябрь 1994)	—	проприетарная	серверы, рабочие станции
Tru64 UNIX (OSF/1 AXP, Digital UNIX)	DEC, Compaq, HP	1992	4.3BSD, Mach 2.5, UNIX System V	5.1B-3 (июнь 2005)	—	проприетарная	серверы, рабочие станции
Mac OS X	Apple Computer	март 2001	NeXTSTEP, FreeBSD, Mac OS	10.4.6 "Tiger" (3 апреля 2006)	129/499 (desktop /server)	Open source core system (APSL, GPL, others) with proprietary higher level API layers	серверы, рабочие станции, домашние десктопы
DragonFly BSD	Matt Dillon	12 июля 2004	FreeBSD 4.8	1.4 (7 января 2006)	free	BSD	серверы, кластеры
FireflyBSD	Steven David Rhodus	14 сентября 2004	DragonFly BSD	1.4	$12.95	—	коммерческая версия DragonFly
PC-BSD	Kris Moore, Mike Albert, Tim McCormick, Dimitri Tishchenko	?	FreeBSD	1.0 (29 апреля 2006)	free	BSD	компьютеры для домохозяек
DesktopBSD	Peter Hofer, Daniel Seuffert	25 июля 2005	FreeBSD	1.0 (28 марта 2006)	free	BSD	компьютеры для домохозяек
BSDeviant3	Unixpunx	?	FreeBSD	(июнь 2004)	free	—	LiveCD
ClosedBSD	various contributors	?	FreeBSD	1.0B(floppy), 1.0-RC1(CD)	free	BSD	firewall/NAT, boot floppy, LiveCD
PicoBSD	Andrzej Bialeck	?	FreeBSD	0.42	free	BSD	boot floppy
MicroBSD 3	Bulgarians	?	OpenBSD 3.0/3.4	0.6 (27 октября 2003)	free	—	малые серверы
Gentoo/ FreeBSD	Gentoo Linux developers	?	FreeBSD	6.1 (9 мая 2006)	free	GPL, BSD	рабочие станции

чество оригинального UNIX-кода с каждой версией все уменьшалось и уменьшалось.

Возникла идея — отделить код, написанный вне AT&T (к которому, главным образом, относился TCP/IP-стек), а код T&T переписать и распространять под открытой лицензией. Так зародился проект Net (не путать с NetBSD), выпустивший две версии: Net/1 и Net/2. Причем Net/2 была перенесена Билом Джолизом на 386 (386BSD), а потом... внезапно грянул гром, сгустились тучи и наступили трудные времена.

Подразделение фирмы AT&T с громким названием Unix System Laboratories в 1994 году подало иск о нарушении авторских прав. Суд рассмотрел дело и пришел к заключению, что из 18000 файлов, входящих в BSD, только 3 файла должны быть изъяты и еще 70 модифицированы, чтобы показывать USL-копирайт. Поэтому новая (и последняя) версия BSD вышла в двух вариантах: 4.4BSD-lite — свободно распространяемая, но без части ключевых файлов, и 4.4BSD-Encumbered — в полном составе, но требующая лицензии от AT&T.

Группа CSRG была распущена, но вместо того, чтобы умереть, 4.4BSD породила множество клонов, доживших до наших дней и занимающих солидную нишу на рынке серверов и высокопроизводительных рабочих станций.

➔ **FreeBSD**. Проект FreeBSD, возглавляемый Джорданом Хабардом, Нэтом Вильямсом и Родом Гримесом, стартовал в начале 1993 года, отпочковавшись от проекта «Unofficial 386BSD Patchkit». Он представлял собой выполненный Биллом Джолицем порт BSD на 386 машине, так и не доведенный до конца. Реализованный им patchkit-механизм создавал много проблем, количество которых увеличивалось с каждым днем, делая работу с системой все более неудобной. Указанная троица активистов предложила Биллу свою помощь, но была отвергнута без каких-либо объяснений. Благо лицензия позволяла дорабатывать систему без его согласия.

Объединив 4.3BSD-Lite («Net/2») с 386BSD и подключив Free Software Foundation, Джордан, Нэт и Род к концу 1993 года сотворили полноценный дистрибутив операционной системы, получивший название FreeBSD (предложенное Дэвидом Гринманом), подчеркивающее свободу использования. Другим важным шагом стало распространение системы на CD-ROM фирмой Walnut Creek, что для пользователей, лишенных интернета (а в 1993 году доступ к нему имели немногие), было очень даже актуально.

Тем временем начался очередной виток судебных разборок вокруг 4.3BSD-Lite, изымающий все новые куски критического кода, что задержало выход FreeBSD 2.0, выпущенной только в конце 1994 года и уже полностью свободной от нападок правообладателей оригинального UNIX-кода, преемником которого стала Novell.

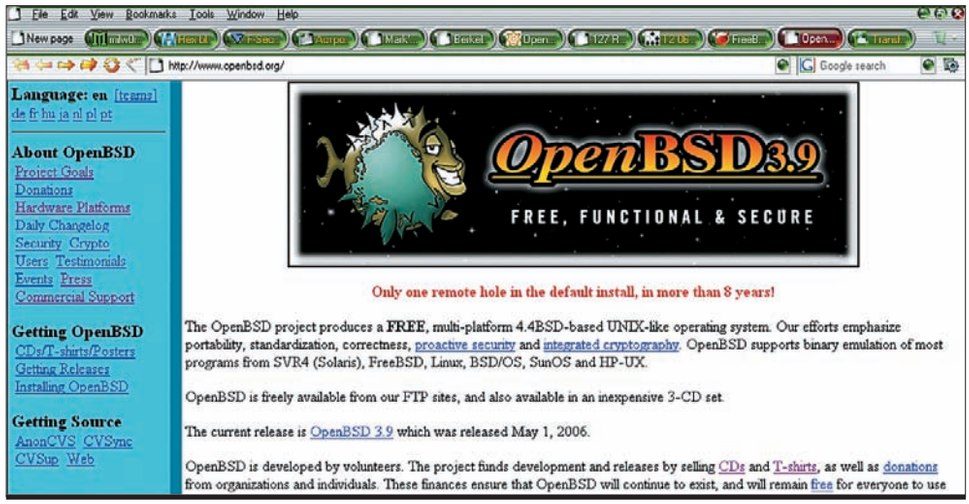
На данный момент текущая версия — 6.1 — придерживает главным образом x86 и другие платформы (Pentium/Athlon/x64-86/UltraSPARCR/IA-64/ARM) и остается самой популярной xBSD-системой. Несмотря на то, что она в основном ориентирована на серверное использование, и, в отличие от LINUX, разработчики FreeBSD не покушаются на рынок десктопов, она используется и там. Кстати говоря, FreeBSD — единственная xBSD-система, для которой фирма NVIDIA периодически выпускает драйвера.

BSD-войны

В КОНЦЕ 2005 ГОДА ГРУППА СЕРТИФИЦИРОВАНИЯ BSD (BSD CERTIFICATION GROUP) ПРОВЕЛА ОПРОС СРЕДИ 4330 ПОЛЬЗОВАТЕЛЕЙ BSD-СИСТЕМ, С ЦЕЛЬЮ СОСТАВЛЕНИЯ РЕЙТИНГА ПОПУЛЯРНОСТИ. ВЫЯСНИЛОСЬ, ЧТО 77% РЕСПОНДЕНТОВ ПРЕДПОЧИТАЮТ FREEBSD, 33% — OPENBSD, 16% — NETBSD, 2,6% — DRAGONFLY И 6,6% ИСПОЛЬЗУЮТ ДРУГИЕ КЛОНЫ BSD

Таблица 2 технические характеристики xBSD-систем

	архитектура	файловая система	тип ядра	GUI	менеджер package'й	менеджер обновлений	основное API
FreeBSD	x86, AMD64, PC98, UltraSPARC, другие	UFS, UFS2, ext2, FAT, ISO 9660, UDF, NFS, SMBFS, NTFS (read only), ReiserFS (read only), XFS (эксперимент.), другие	монолитное	нет	ports tree, packages	source (CVSup, portsnap), network binary update (freebsdupdate)	BSD, POSIX
OpenBSD	x86, 68k, Alpha, AMD64, SPARC, VAX, другие	UFS, ext2, FAT, ISO 9660, NFS, NTFS7 (read only), AFS, others	монолитное	нет	ports tree, packages	source (CVS, CVSup, rsync) or binary upgrade	BSD, POSIX, X11
NetBSD	x86, 68k, Alpha, AMD64, SPARC, VAX, другие	UFS, UFS2, ext2, FAT, NFS, LFS, другие	монолитное	нет	pkgsrc	source (CVS, CVSup, rsync) or binary (using sysinst)	BSD, POSIX
Mac OS X	PPC, x86	HFS+ (по умолч.), HFS, UFS, AFP, ISO 9660, FAT, UDF, NFS, SMBFS, NTFS (read only), FTP, WebDAV, другие	гибридное	есть (Aqua)	OS X Installer	Software Update	Carbon, Cocoa, BSD/POSIX, CF, X11 (since 10.3)
DragonFly BSD	x86	UFS, FAT, ISO 9660, NFS, SMBFS, NTFS (read only), другие	гибридное	нет	pkgsrc, ports tree	CVSup	BSD, POSIX
PC-BSD	x86, AMD64	UFS, UFS2, FAT, ISO 9660, NFS, SMBFS, NTFS (read only), другие	монолитное	есть (KDE)	graphical installation wizard, ports tree	CVSup, Portsnap, network binary update (Online Update)	BSD, POSIX, X11, KDE



За 8 лет только одна удаленная дыра в конфигурации по умолчанию

Но сравнивать LINUX и FreeBSD некорректно хотя бы потому, что в LINUX разработкой ядра занимается один коллектив, а дистрибутивы клепают все кому не лень, что порождает несовместимость и неразбериху. А во FreeBSD и ядро, и прикладные программы находятся в одном CVS. Продукты сторонних производителей с закрытым кодом включаются в дистрибутив только при необходимости (например, драйвера). Но все-таки включаются, что, в конечном счете, идет на благо пользователей.

Лицензия BSD относится к числу наиболее демократичных и, в отличие от GPL, являющейся прототипом «развитого социализма», действительно предоставляет полную свободу в использовании исходного кода, в том числе и закрытых коммерческих продуктов, таких как CISCO OS, MAC OS X, Windows и т. д. Демократичность проявляется и в отношение главного конкурента — FreeBSD: BSD поддерживает режим эмуляции LINUX (Linux compatibility layer), позволяя запускать двоичные программы, исходные тексты которых недоступны: StarOffice, Netscape, Adobe Acrobat, RealPlayer, VMware, Oracle, WordPerfect, Skype, Doom 3, Quake 4, Unreal Tournament, SeaMonkey и т. д.

FreeBSD сохранила чертенка Бистли в качестве логотипа, но в 2005 году объявила конкурс на его «стилизованную» версию, победителем которого стало изображение «рогатой» сферы.

➔ **NetBSD.** За полгода до появления проекта FreeBSD четыре программиста (Крис Деметру, Тео де Раадт, Дама Гласс и Чарлз Ханнум) решили переработать исходные тексты BSD, чтобы повысить качество кода и максимально упростить его перенос на другие архитектуры, включая процессоры с ограниченными возможностями, используемые во встраиваемых системах.

Желая подчеркнуть сетевую ориентацию будущей системы, ее назвали NetBSD и, отталкиваясь от 4.3BSD, Net/2 и 386BSD, программисты уже в начале 1993 года выпустили первый официальный релиз — NetBSD 0.8, работающий только на PC и «потерявший» несколько утилит из Net/2.

Логотипом системы стал флаг, поднимаемый чертями, попирающими PC. Выглядело слишком задиристо, и в следующей «ревизии» черти и компьютеры из логотипа были изъяты. Остался лишь стилизованный флаг.

Следующий релиз — NetBSD 0.9, вышедший через полгода, в основном представлял собой «работу над ошибками» без существенных улучшений. NetBSD 1.0, вышедшая в конце 1994 года, оказалась первой многоплатформенной Net-системой, поддерживающей, помимо x86, еще и HP 9000 Series 300, Amiga, 68k Macintosh, Sun-4c и PC532. Как и FreeBSD, NetBSD столкнулась с необходимостью переписывания больших кусков изъятого кода, что задержало выпуск следующей версии.

Но процессорные платформы умирали быстрее, чем их успевали поддерживать. Архитектура x86 стремительно захватывала все новые рыночные ниши, и легкость переноса уже не являлась решающим фактором, определяющим популярность системы. Пользователей гораздо больше интересовала стабильность и защищенность. В результате среди разработчиков возник раскол, закончившийся в 1994 году выходом Тео де Раадта из проекта. Но, несмотря на это, развитие NetBSD было продолжено.

В конце 2004 вышла NetBSD 2.0, а еще через год — NetBSD 3.0, поддерживающая свыше полусотни различных платформ (в том числе и PDA), естественную (native) многопоточность, симметричные многопроцессорные системы... Но особой популярности она так и не смогла завоевать.

➔ **OpenBSD.** В 1994 году Тео де Раадт покинул лагерь NetBSD с тем, чтобы в конце 1995 года основать свой клон NetBSD, концентрирующийся на защищенности и безопасности. Руководствуясь лозунгом «security by default» (безопасность по умолчанию), Тео де Раадт вместе с единомышленниками кардинально переработали конфигурацию системы, действуя по принципу «все, что явно не разрешено — запрещено», и провели тщательный аудит исходных текстов на предмет всевозможных ляпов и дыр, которых там было предостаточно.

Новая система получила название OpenBSD, с логотипом, изображающим ощетинившуюся рыбу-собаку (pufferfish), с колючками, выпирающими во все стороны, что, по всей видимости, символизировало «хрен меня сломаешь». Впрочем, также использовался и чертенок (с нимбом над головой), а рыба-собака со временем обрела снисходительную улыбку.

Первая публичная версия OpenBSD 1.2 вышла в середине 1996 года, а через несколько месяцев появилась и OpenBSD 2.0. Воспользовавшись сетевым сканером Ballista (позже переименованным в Cybercop Scanner), созданным по спецзаказу компанией Secure Networks, разработчики продолжали вылавливать потенциальные уязвимости и усиливать защищенность, выпустив через некоторое время OpenBSD 2.3.

Политика включения в дистрибутив постороннего кода очень жестокая, и если производитель «зажимает» исходные тексты, делая невозможным их аудит, то этот код не включается, несмотря на потери конечных пользователей.

Система действительно оказалась более устойчивой к атакам, чем ее клоны, и вплоть до июня 2002 года на **www.openbsd.org** красовался слоган «No remote computer hole in the default install, in nearly 6 years» (ни одной удаленной дыры в конфигурации по умолчанию за последние 6 лет). Но затем Марк Давд из Internet Security Systems обнаружил уязвимость в OpenSSH (**http://xforce.iss.net/xforce/alerts/id/advise123**), позволяющую атакующему заполучить права root'a. Поэтому слоган пришлось менять, и в настоящее время на **www.OpenBSD.org** значится «Only one remote hole in the default install, in more than 8 years!» (только одна удаленная дыра в конфигурации по умолчанию за более чем восьмилетний срок!).

В середине 2006 года была выпущена последняя на данный момент версия OpenBSD 3.9, остающаяся самой защищенной и широко используемой в качестве серверов в критических инфраструктурах BSD-системой, хотя и не так широко, как FreeBSD.

➔ **заключение.** Мир xBSD-систем довольно разнообразен и достаточно дружелюбно настроен к профессионалам. В отличие от LINUX, технология которого тесно смешена с пропагандой, а конструктивные огрехи затыкаются идеологической подоплекой противостояния Microsoft, xBSD крепко держит свою рыночную нишу и никаким «миссионерством» не занимается, поскольку «объять необъятное нельзя». И пока LINUX стремительно превращается в Windows, перенимая ее худшие черты и теряя свои преимущества, высоко ценимые профессионалами (в первую очередь — предсказуемость поведения и командную строку), BSD с годами только развиваестя, становясь все крепче на ноги и обеспечивая удобство и комфорт тем, кто разбирается в этом **C**

бессмертный BSD

ОБЗОР И СОЗДАНИЕ LIVECD FREEBSD

LIVECD — ВЕСЬМА ПОПУЛЯРНАЯ ВЕЩЬ В СОВРЕМЕННОМ МИРЕ. ПРАКТИЧЕСКИ КАЖДЫЙ LINUX-ДИСТРИБУТИВ ОБЗАВОДИТСЯ LIVECD-ВЕРСИЕЙ, ДА И ОТДЕЛЬНЫХ «ЖИВЫХ ДИСКОВ» ХВАТАЕТ (ПОИСК ПО DISTROWATCH ДАЕТ НАМ БОЛЕЕ 170 НАИМЕНОВАНИЙ). А КАК ЖЕ ОБСТОИТ ДЕЛО С LIVECD-ДИСТРИБУТИВАМИ НА ОСНОВЕ XBSD?

МОЖАЙСКИЙ СЕРГЕЙ
{TECHNIX@FRENZY.ORG.UA}

Естественно, такие дистрибутивы имеются. Их пока еще немного, но они активно развиваются. В этой статье я расскажу о существующих LiveCD на основе BSD-систем, об их внутреннем устройстве и о том, как создать такие LiveCD самостоятельно. Ну что, приступим?

➔ **FreeBSD LiveCD.** Первый LiveCD на основе ОС FreeBSD был создан в 2001 году Бразильской группой пользователей FreeBSD. Назывался он весьма незатейливо — «FreeBSD LiveCD». Его разработка была прекращена в 2002 году. Та же судьба в разное время постигла еще несколько проектов (BSDeviant, LiveBSD, Snarl). В наши дни единственными развивающимися LiveCD на базе FreeBSD являются итальянский FreeSBIE и украинский Frenzy.

FREESBI

www.freesbie.org
Размер: 610 Мб
Версия: 1.1

Созданием этого дистрибутива занимается итальянская группа пользователей FreeBSD.

Основными задачами проекта FreeSBIE являются разработка набора программ для создания собственных CD и создание набора готовых ISO-образов для различных задач.

На данный момент командой FreeSBIE выпущен только один вариант готового LiveCD, предназначенного для тех, кто желает познакомиться с FreeBSD или использовать ее в качестве своей рабочей операционной системы.

После загрузки системы нужно выбрать предпочительную раскладку клавиатуры для консоли и иксов, а также графическую оболочку. В качестве рабочего окружения предлагаются на выбор XFce или fluxbox. Набор софта традиционен для подобных LiveCD — офис, браузер, почтовый клиент и различные мультимедийные приложения. FreeSBIE можно установить на жесткий диск, для этого в комплект входит программа установки, основанная на BSDinstaller. Имеется также возможность сохранить настройки на любой смонтированный диск. На диске можно найти краткую (очень краткую) документацию по системе на английском языке.

К сожалению, с поддержкой русского языка дела совсем плохи. Несмотря на то, что русский язык можно выбрать из списка, русских букв в FreeSBIE ты не увидишь по причине отсутствия русских шрифтов. Более того, ты не сможешь даже использовать английский язык, так как в конфиге иксов устанавливается только одна раскладка клавиатуры. В общем, засада.

Единственное, чем выделяется этот дистрибутив, так это отличными сборочными скриптами. Но о них чуть позже.

FRENZY

www.http://frenzy.org.ua/
Размер: 50-250 Мб
Версия: 1.0

Задачей проекта Frenzy является создание удобного инструмента для системных администраторов. Последняя версия, Frenzy 1.0 (Dreamchild) вышла совсем недавно, в июне этого года. Выпускается она в трех версиях: Frenzy standard для системных администраторов (200 Мб), Frenzy extended — для админов и опытных пользователей (250 Мб) и Frenzy lite с консольными утилитами (50 Мб).

Всего в состав дистрибутива входит более 500 приложений, в качестве графической оболочки используется Fluxbox.

Иксы по умолчанию не стартуют — их можно запустить командой «startx», либо указав опцию «gui» при загрузке системы.

Набор программ весьма разнообразен, но разобраться в этом ворохе софта очень просто — в меню Fluxbox он рассортирован по темам. В общем, админу или хакеру подборка софта очень понравится.

Помимо традиционного sysinstall, в Frenzy присутствует утилита конфигурации frconf для настройки системы. Сетевой конфигуратор netconf поможет настроить локальную сеть, модемное соединение, а также VPN и ADSL-подключение, а с помощью конфигурационной утилиты serconf можно быстро поднять ssh-сервер или сделать из Frenzy web-сервер, ftp-сервер и сервер Samba.

Настройки можно сохранить с помощью утилиты frbk на дискету, жесткий диск или USB Flash, при следующей загрузке они будут автоматически восстановлены. Более того, в Frenzy FAQ описан метод включения сохраненных настроек в ISO-образ — для этого достаточно открыть исошник с помощью программы UltraISO и добавить архив с настройками в папку frenzy/backup.

Frenzy 1.0 можно установить на жесткий диск или USB Flash с помощью простой программы-инсталлятора. Правда, простота не обошлась даром — при установке на жесткий диск вся система устанавливается на один слайс, и создать отдельные разделы для /var, /usr и т.п. не выйдет. Так что ставить Frenzy на HDD в качестве сервера я бы не стал :). После установки Frenzy на жесткий диск ты получишь практически полноценную ОС FreeBSD с отличным набором программ. Останется только скачать и установить исходные коды системы и коллекцию портов.

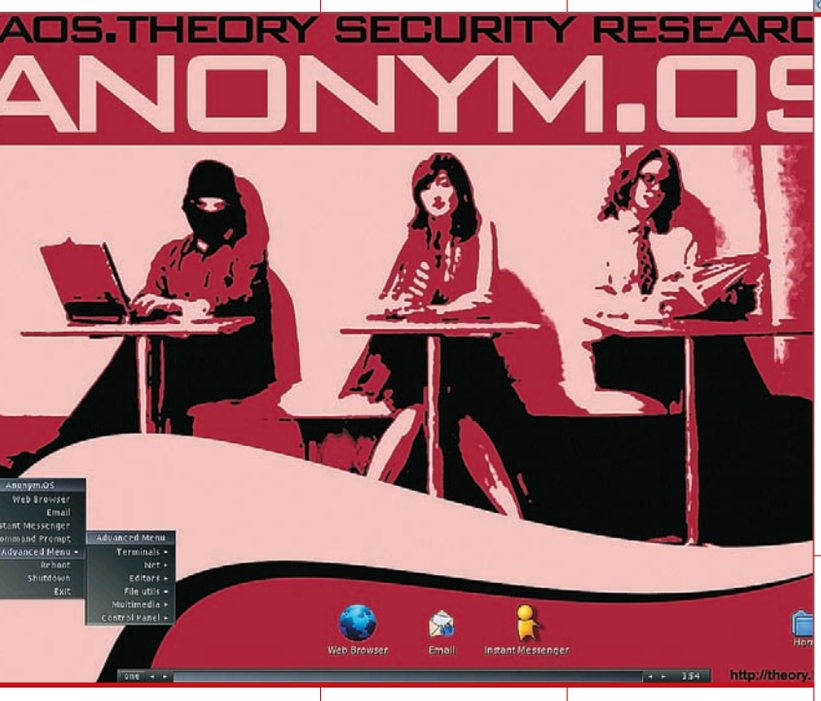
Что касается установки на флешку, стоит отметить, что утилита установки разбивает ее на два раздела: FAT32 для хранения данных и UFS2 для Frenzy, так что после установки Flash-диск вполне можно использовать для хранения данных.

Немаловажным достоинством Frenzy является качественная русификация системы — проблем с русским ни в консоли, ни в графической оболочке не возникает. Да и подробная документация на русском языке присутствует. В общем, must have.

ANONYM.OS

http://sourceforge.net/projects/anonym-os/
Размер: 550 Мб
Версия: ShmooCon 2006

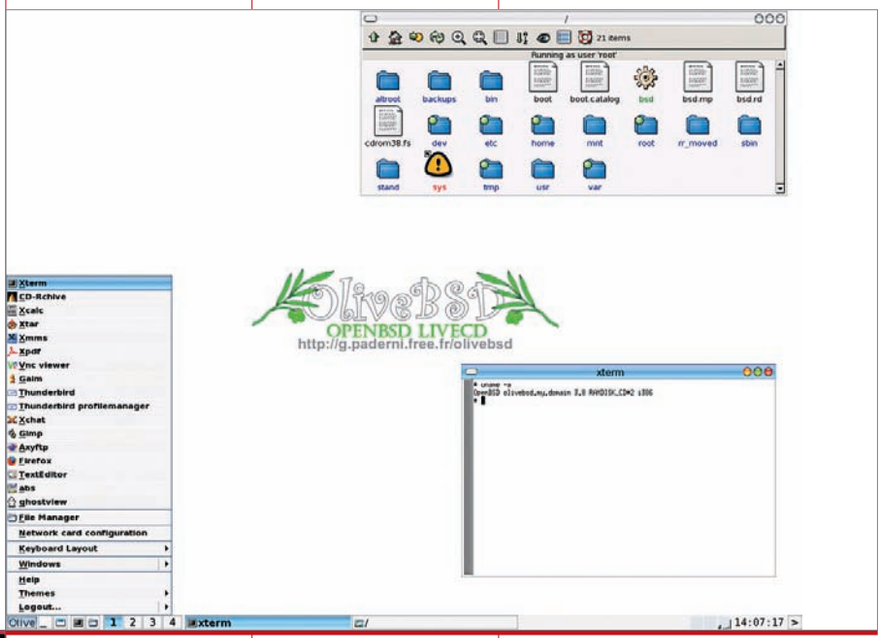
Этот дистрибутив был разработан security-командой kaos.theory и представлен общественности на хакерской конференции Shmoo Con в феврале 2006 года. Основная задача Anonym.OS — обеспечить безопасные и анонимные вебсерфинг, переписку и общение. Набор программ соответствует назначению — кроме Firefox, Thunderbird и Gaim, в состав Anonym.OS входят анонимайзер Tor и прокси-сервер Privoxy. Свою задачу по обеспечению анонимности Anonym.OS выполняет отлично (стоит отметить, что он прикидывается Windows XP SP1, если попытаться просканить его nmap'ом), но, кроме работы в инете, дистрибутив больше ни для чего не пригоден, да и с русским языком он совсем не дружит.



OLIVEBSD

http://g.paderni.free.fr/olivebsd/
Размер: 700 Mb
Версия: нет

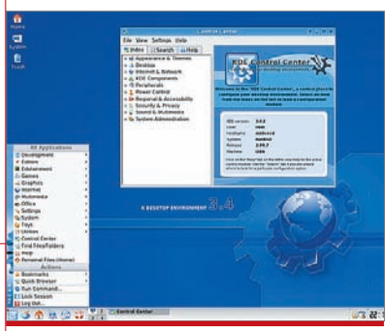
Я так понимаю, что назначение этого дистрибутива — показать, что и на OpenBSD можно сделать нормальный LiveCD :). Что ж, разработчику это удалось! OliveBSD основан на OpenBSD 3.8 и использует в качестве графической оболочки IceWM. Набор программ невелик, однако покрывает все потребности обычного пользователя. Многие пользователи отмечают, что OliveBSD отказывается загружаться на некоторых компьютерах, а это совсем не радует. В общем, как advocacy tool дистрибутив весьма полезен, но для реальной работы лучше поискать что-то еще.



NETBSD LIVE!

ftp://ftp.netbsd.org/pub/NetBSD/misc/xtraeme
Размер: 646 Мб
Версия: 3.99.7

NetBSD'шный LiveCD, увы, не имеет своего сайта и просто лежит себе на официальном FTP. Так что единственным кладезем информации стал лежащий там же файлик README.LIVECD. Итак, этот LiveCD позволяет нам попробовать поработать с NetBSD, не устанавливая ее на жесткий диск. В качестве графической среды нам предлагается KDE и стандартный набор программ для него — для пользователя этого вполне достаточно. Так что свою задачу — познакомить людей с NetBSD — дистрибутив успешно выполняет.



```
>>> stage 2.1: cleaning up the object tree
>>> stage 2.2: rebuilding the object tree
>>> stage 2.3: build tools
>>> stage 3.1: making dependencies
>>> stage 3.2: building everything
>>> Kernel build for FREESBIE completed on Mon Jul  3 13:24:01 EEST 2006
#### Installing world for i386 architecture ####
>>> Making hierarchy
>>> Installing everything
#### Installing kernel for i386 architecture ####
>>> Installing kernel
#### Cloning /usr/local/freesbie-fs to /usr/local/freesbie-clone ####
Compressing /usr/local/freesbie-clone/uzip/usr.ufs... 66.2194% saved
#### Building bootable ISO image for i386 ####
Saving mtree structure...
Running mkisofs...
ISO created:
```

Процесс сборки FreeSBIE

→ **OpenBSD и NetBSD LiveCD.** Тут количество LiveCD заметно меньше. Кроме Anonym.OS, OliveBSD и NetBSD live!, единственным кандидатом на включение в обзор мог бы стать NetBSD'шный Newbie (**http://arudius.sourceforge.net/**), но он находится на стадии зародыша.

Основная проблема LiveCD на базе Open-BSD и NetBSD — отсутствие поддержки сжатой файловой системы, поэтому они показывают весьма низкую скорость загрузки программ. Наберись терпения.

→ **как устроены LiveCD.** Создание LiveCD — теоретически, весьма несложный процесс. Первое, что приходит в голову — создать на CD точную копию файловой системы, указать в качестве загрузчика cdboot и немного подправить стартовые скрипты. По этому принципу построены почти все BSD'шные LiveCD.

Однако у этого способа есть большой недостаток. Скорость работы такого LiveCD будет крайне низкой, в 3-5 раз ниже, чем у линуксовых аналогов, что нам и демонстрируют LiveCD на основе NetBSD и OpenBSD. Замедление происходит из-за того, что много времени тратится на чтение данных с компакта, а уж если они разбросаны по разным частям CD, то загрузка программы осуществляется мучительно долго. Использование сжатой файловой системы улучшает ситуацию: объем данных, читаемых с компакта для запуска той или иной программы, сокращается в несколько раз, а распаковка этих данных на современных компьютерах происходит почти мгновенно, и скорость работы LiveCD существенно повышается.

В настоящее время сжатая файловая система есть только в FreeBSD. За ее поддержку отвечает модуль geom_uzip, написанный Максимом Хоном. По функциональности он аналогичен cloop из KNOPPIX и совместим с ним, но использует совсем другой механизм работы с файловой системой. Сжатие, предоставляемое geom_uzip, составляет порядка 65%.

Первым дистрибутивом с использованием geom_uzip стал FreeSBIE. В нем использовалась комбинированная схема — корневая файловая система находилась на CD, а файловые системы /usr, /usr/X11R6 и /var подключались из сжатого образа.

В Frenzy использован совершенно другой подход. После загрузки ядра в качестве корневой файловой системы подключается небольшой образ файловой системы с минимумом утилит, необходимым для поиска компакт-диска, его монтирования и подключения сжатой файловой системы из файла на нем. Вся файловая система Frenzy находится в одном файле frenzy.uzip, что позволило реализовать такие возможности, как полная загрузка в оперативную память и загрузка из образа на жестком диске. Следующая задача, стоящая перед разработчиками LiveCD — сделать так, чтобы система, загруженная с диска, могла куда-то записывать файлы и чтобы некоторые файлы на CD можно было создавать и редактировать — в общем, имитировать работу с жесткого диска. В большинстве линуксовых дистрибутивов для этого используется unionfs — с ее помощью можно сымитировать, что весь CD доступен для записи, смонтировав диск в оперативной памяти поверх файловой системы на CD.

Но, увы, в FreeBSD штатный unionfs пока что в плачевном состоянии и почти непригоден к повседневному использованию. Оздоровлением ситуации сейчас занимается японец Даичи Гото, и, судя по отзывам, новый unionfs уже сейчас показывает отличные результаты. Так что приходится использовать обходные пути. В FreeSBIE, основанной на FreeBSD 5.3, используется unionfs, но только для подключения /etc, /var и /root. В Frenzy, где базовой системой является FreeBSD 6.1, использовать unionfs было бесполезно, поэтому там использовано mount_nullfs, с помощью которого поверх оригинальных /etc, /root, /var монтируются файловые системы с теми же файлами, но уже с RAM-диска.

Для того чтобы LiveCD корректно работал на любом компьютере, он должен определять состав аппаратного обеспечения компа для обеспечения поддержки всех девайсов. Вопрос с определением оборудования решается разработчиками BSD'шных LiveCD «в лоб» — в ядро вкомпилирована поддержка практически всего оборудования, поддерживаемого системой. Это ускоряет загрузку, но в то же время увеличивает размер памяти, занимаемой ядром. Правда, разработчики FreeSBIE решили выделиться и сделали скрипт определения звуковой карты с загрузкой соответствующего модуля ядра.

В FreeSBIE и Frenzy есть утилиты для автоматического поиска и монтирования разделов винчестера. В Frenzy это происходит автоматически при загрузке (по умолчанию разделы монтируются в режиме «только для чтения», но можно выбрать и режим «для записи» или вообще отказаться от монтирования разделов), а в FreeSBIE скрипт монтирования нужно запускать вручную.

→ **делаем LiveCD сами.** Использовать готовые LiveCD — это, конечно, приятно и полезно. Но что

Сравнение различных liveCD

	FreeSBIE	Frenzy	OliveBS	Anonym.OS	NetBSD Live!
Базовая ОС	FreeBSD	FreeBSD	OpenBSD	OpenBSD	NetBSD
Размер ISO, Мб	610	50-250	700	550	646
Сжатая FS	•	•	-	-	-
Установка на HDD	•	•	-	-	-
Установка на USB Flash	-	•	-	-	-
Полная загрузка в RAM	-	•	-	-	-
Сохранение настроек	•	•	-	-	-
Наличие документации	•	•	-	-	-
Поддержка русского языка	-	•	-	-	-
Офисные приложения	•	-	•	-	-
Браузер, почта, IM	•	•	•	•	•
Мультимедиа	•	•	•	-	•
Сетевые утилиты	•	•	•	•	•
Системные утилиты	-	•	-	-	-
Security-утилиты	-	•	-	•	-
Наличие сборочных скриптов	•	•	-	-	-

делать, если хочется создать свой единственный и неповторимый LiveCD на основе любимой тобой BSD-оси?

Поклонникам NetBSD далеко ходить не надо: LiveCD можно создать с помощью пакета sysutils/mklivecd из коллекции pkgsrc. А вот для OpenBSD готовых решений вообще не существует, немного информации по сборке LiveCD на опенке можно найти по этим адресам: **www.onlamp.com/pub/a/bsd/2005/07/14/openbsd_live.html и www.blackant.net/other/docs/howto-bootable-cdrom-openbsd.php.**

Зато для фришников есть целых два набора сборочных скриптов — FreeSBIE и Frenzy SDK. Вот с ними мы и познакомимся более внимательно.

Скрипты FreeSBIE заслуживают наивысшей похвалы. Текущая их версия, названная FreeSBIE2, написана Даниэлем Френи в рамках проекта Google Summer of Code и позволяет собирать LiveCD для архитектур i386, amd64 и powerpc. В основе этих скриптов лежат традиционные Makefile и конфигурационные файлы. Самой мощной фичей является модульная структура, позволяющая добавлять в сборочную инфраструктуру свои модули для поддержки различных дополнительных возможностей (к примеру, установки на HDD, определения оборудования и т.п.).

Свежую версию скриптов FreeSBIE можно получить через CVS:

```
cvs -d :pserver:anonymous@cvs.freesbie.org:/cvs login
cvs -d :pserver:anonymous@cvs.freesbie.org:/cvs co freesbie2
```

Перед началом сборки запускаем команду

```
make pkgselect
```

для выбора пакетов, которые будут включены в создаваемый LiveCD. Пакеты выбираются из тех, что уже есть в системе. Теперь, если мы хотим что-то изменить в параметрах сборки, идем в каталог conf и копируем файл freesbie.defaults.conf в freesbie.conf, после чего меняем значения переменных уже в нем. Особенно полезен для нас параметр EXTRA, позволяющий указать список плагинов, используемых при сборке LiveCD. Сами плагины лежат в каталоге extra.

Итак, мы сделали все нужные настройки, и все готово к сборке. Переходим в основной каталог скриптов и набираем:

```
make iso
```

Разработчики рекомендуют посмотреть кино, пока идет сборка. Занимает она несколько часов, так что советом стоит воспользоваться. После окончания сборки готовый исошник будет лежать в /usr/obj.

В отличие от FreeSBIE, сборочные скрипты Frenzy не так универсальны, поскольку оптимизированы для построения одного-единственного дистрибутива.

Так что собрать дистрибутив будет посложнее.

Для начала нам нужно скачать сам SDK. Полный SDK занимает около гигабайта. Найти его можно на разных ftp-миррорах Frenzy, например на **ftp://ftp.frenzy.org.ua/pub/Frenzy/1.0/sdk/.** Создаем chroot-окружение:

```
make buildworld
make installworld DESTDIR=/usr/Frenzy
```

Закидываем туда дерево исходных кодов (/usr/Frenzy/usr/src), дерево портов (если будем собирать какие-то дополнительные пакаджи, /usr/Frenzy/usr/ports) и сами сборочные скрипты (/usr/Frenzy/buildscripts).

Теперь подредактируем файл config, находящийся в каталоге со сборочными скриптами.

Для нас важны параметры FRENZY_TYPE (может принимать значения std или ext) и FRENZY_LANG (ru или en).

Пакаджи нужно собрать и положить в нужные каталоги заранее. Пакаджи, собранные специально для Frenzy, скачиваем из SDK (файл Frenzy.tar) и складываем в buildscripts/packages/Frenzy. Обычные фришные пакаджи нужно сложить в buildscripts/packages/FreeBSD.std или в buildscripts/packages/FreeBSD.ext, в зависимости от выбранного типа сборки. Можно просто скачать с ftp те пакаджи, которые были собраны для имеющейся версии Frenzy, выбрать софт из них и положить нужные пакаджи в каталог.

Наконец, все готово. Переходим в наше chroot-окружение и начинаем сборку:

```
chroot /usr/Frenzy mount -t devfs devfs /dev
chroot /usr/Frenzy /bin/tcsh
cd buildscripts
./frbuild all
```

После сборки готовый исошник будет лежать в каталоге ISO.

Подробнее о сборке Frenzy можно прочесть в документации по Frenzy SDK(**http://frenzy.org.ua/ru/releases/1.0/doc/doc-sdk.html**) C

SPECIAL ОБЗОР

HARD

ОПТИМИЗАЦИЯ ПРОИЗВОДИТЕЛЬНОСТИ UNIX

М.: Альфа-букс, 2002
Маджидимер А.
465 страниц
Разумная цена: 129 р.

Редко можно встретить человека, который детально знал бы систему и мог бы ее также детально настроить. Так как это требует хороших знаний компьютерной архитектуры, устройства системы UNIX и средств мониторинга ее производительности. В книге автор освещает вопросы оптимизации, рассматривая цепочку проблем от пользовательских приложений до технических средств. Изначально предполагая, что ты не имеешь достаточно глубоких знаний ни в области архитектуры компьютерных систем, ни в области внутренней организации UNIX. Причем большая часть книги посвящена именно анализу, а не готовым рецептам. Автор считает, что если ты научишься грамотно анализировать, то сможешь решить любую даже самую сложную проблему.

MEDIUM

ОТ WINDOWS К LINUX

М.: ООО «Бином-Пресс», 2005 / Марсель Гане
336 страниц
Разумная цена: 223 р.

Книга, что называется, для простых пользователей, которые решили попробовать себя в Linux. Без необходимости что-то ставить, разбираться, настраивать... Все это доступно благодаря MandrakeMove — дистрибутиву Linux, основанному на версии Mandrake 10. Операционная система размещена на прилагаемом к книге диске и загружается с него полностью. Умная система самостоятельно определит параметры твоего компьютера. Естественно, это несколько медленнее, чем если бы ты использовал Linux, установленный на жесткий диск. Зато не придется деинсталлировать Windows или другие установленные системы. Плюс ты ограничен программными пакетами, которые присутствуют на диске. Но, если вдруг тебя попрет от Linux, ничто не мешает тебе уже основательно поставить его на жесткий диск, о чем также сказано в книге.

подземелья мудрости

КНИГОХРАНИЛИЩА BSD-ЗНАНИЙ В ИНТЕРНЕТЕ

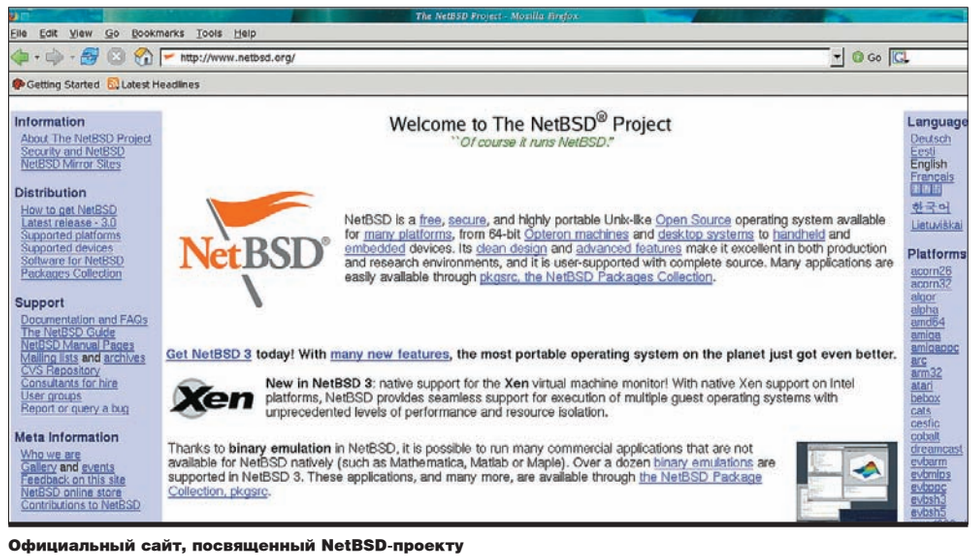
КТО УМЕН, А КТО ДУРАК, КТО ЗА КНИГУ — КТО В КАБАК, КАК НАС УЧИТ КОММУНИСТИЧЕСКАЯ МУДРОСТЬ. НО МЫ, ЛЮДИ XXI ВЕКА, ЗА ЗНАНИЯМИ ДВИГАЕМСЯ ИСКЛЮЧИТЕЛЬНО В СЕТЬ.

WOLF D.A. AKA PAYHASH

→ **BSD**. В конце 70-х годов прошлого столетия ОС UNIX System V6 была серьезно недоработана. Билл Джой, кернел-хакер из Беркли, сделав небольшие изменения в коде ядра, пришел к выводу, что одними исправлениями здесь не обойтись, и занялся разработкой собственного дистрибутива. Так, 9 марта 1978, года появился первый релиз операционной системы Беркли — Berkeley Software Distribution. Он включал в себя Pascal-систему со всеми исходными текстами и текстовый редактор ex. В течение следующего года по разным вузам разошлось 30 копий новой ОС. Библиотека termcap с поддержкой новых терминалов и великий и ужасный vi послужили причиной создания Second Berkeley Software Distribution, вышедшей 10 мая 1979 года. За последующее десятилетие сменилось множество BSD-версий, в которых появились такие реализации, возможности и программы, как командная оболочка C, управление заданиями, быстрая файловая система Berkeley, надежный механизм сигналов, концепция виртуальной памяти, стек протоколов TCP/IP. Кроме того, во время разработки BSD большое внимание уделялось разделению кода ядра на машинно-зависимые и независимые части, чтобы в дальнейшем было проще производить адаптацию под новые процессоры. FreeBSD, OpenBSD, NetBSD, DragonFlyBSD и даже ядро Mac OS X базируются, в основном, на версии 4.4BSD.

→ **FreeBSD.** Когда речь заходит о представителях BSD-систем, невольно вспоминается FreeBSD и ее логотип — симпатичный чертенок Beastie. FreeBSD — это *nix-подобная свободно распространяемая операционная система для платформ i386, amd64, PC-98, Alpha/AXP и UltraSPARC, которая была разработана на основе 386BSD и 4.4BSD-Lite с некоторыми усовершенствованиями, взятыми из 4.4BSD-Lite2. Оптимизированная для процессоров Intel, быстрая и надежная система не только прочно обосновалась на почтовых и Web-серверах крупнейших компаний и интернет-провайдеров, но и с успехом используется студентами и рядовыми пользователями по всему миру для работы, образования и отдыха.

Исчерпывающую информацию по этой операционке можно найти как на официальном сайте **http://www.freebsd.org/ru/docs.html** , так и на других информационных ресурсах, ссылки на которые можно почерпнуть здесь: **http://www.opennet.ru/links/sml/35.shtml**.



Официальный сайт, посвященный NetBSD-проекту

Официальный сайт, посвященный TrustedBSD-проекту

По FreeBSD довольно много книг на русском, но, пожалуй, хрестоматийной можно назвать книгу с «бабочкой» от Таймэна Брайана и Эбена Майкла. В ней подробно рассказано об инсталляции системы, управлении загрузкой, конфигурировании X-Window и базовом администрировании этой ОС.

➔ **NetBSD.** Первая версия NetBSD, появившаяся в 1993 году, была основана на исходном коде системы 4.3BSD Lite, разработанной университетом Беркли, и системе 386BSD, которая стала первым вариантом BSD Unix, способным работать на процессорах Intel 386. На протяжении своего существования NetBSD впитывает самые лучшие идеи из всех веток BSD-систем. Многие из этих идей постепенно трансформируются и улучшаются энтузиастами, работающими над развитием этого проекта.

Главным козырем NetBSD является многоплатформенность. Эта ОС запускается и работает на всем, где только есть процессор, и даже на кухонных тостерах. i386, amd64, Sun Sparc, HP PA, DEC Alpha, PowerPC, Atari, Commodore Amiga… — этот перечень состоит более чем из 60 архитектур. Вполне возможно, что в будущем под управлением NetBSD будут работать роботы и робототехнические устройства.

Полный набор документации можно найти на официальном сайте **http://www.netbsd.org**. Не так давно в интернете появилась online-книга NetBSD Internals, написанная Julio M. Merino Vidal (**http://www.netbsd.org/Documentation/internals/en**), к сожалению, пока без перевода на русский. К интересным русскоязычным ресурсам, посвященным NetBSD, можно отнести **http://netbsd.webfabrika.ru/** и **http://www.dreamcatcher.ru/** .

➔ **OpenBSD.** В 1995 году произошел раскол в команде core-разработчиков NetBSD. Из-за разногласий по поводу дальнейшего развития этой операционки Тео де Раадт был вынужден покинуть проект. Набрав новую команду энтузиастов, он на базе NetBSD создал свою собственную BSD-систему — OpenBSD, основной упор в которой делается на максимальную защищенность. На сегодняшний день OpenBSD является наиболее безопасной, свободной и лицензионно чистой из всех существующих операционных систем.

Книг, посвященных OpenBSD, совсем немного. Твоему вниманию могу лишь предложить Absolute OpenBSD, Building Firewalls with OpenBSD and PF, 2nd Edition и Secure Architectures with OpenBSD. Сетевых источников информации по OpenBSD крайне мало. Перечислю три наиболее популярных и поддерживаемых из них: **http://www.openbsd.org** — собственно, официальный сайт проекта, **http://undeadly.org** — новостной сайт разработчиков OpenBSD и **http://www.openbsd.ru** — сайт русскоязычных пользователей OpenBSD.

➔ **M0n0BSD**. Особого внимания заслуживает проект MonoWall (**http://m0n0.ch**) — мини-дистрибутив, основанный на FreeBSD. Предназначен для использования на встраиваемых системах, выполняющих роль интернет-шлюза или беспроводной точки доступа. Умеет загружаться с CompactFlash и CD. Сохраняет конфигурацию на флоппи-диске DOS'овского формата в виде одного XML-файла. Все скрипты, включая обработку конфигурации при начальной загрузке, написаны на PHP. Имеет довольно удобное управление через web-интерфейс.

➔ **MacOS X.** Mac OS X — операционная система фирмы Apple Computer, потомок NeXTStep, выпускается для компьютеров Macintosh на базе процессоров PowerPC и Intel. За графическими/мультимедиа возможностями и дружественным интерфейсом далеко не сразу можно уловить BSD-корни. К сожалению, распространяется на коммерческой основе.

➔ **DragonFlyBSD.** Название DragonFly очень четко отражает направление мыслей ее создателя Мэтью Диллона, бывшего разработчика FreeBSD (кстати, его перу принадлежит обновленная система виртуальной памяти FreeBSD. На основе его идей была переписана соответствующая часть ядра Linux, кроме того, он является создателем C-компилятора для AmigaOS и планировщика задач dcron):

[1] «СТРЕКОЗА» — ОДНО ИЗ САМЫХ СОВЕРШЕННЫХ ТВОРЕНИЙ ПРИРОДЫ (ДОСЛОВНЫЙ ПЕРЕВОД).

[2] DRAGON — «ДРАКОН» — СОГЛАСНО КИТАЙСКОЙ МИФОЛОГИИ, СИМВОЛИЗИРУЕТ МУДРОСТЬ.

[3] FLY — «ЛЕТАТЬ» — ЛЕГКОСТЬ, НЕОБРЕМЕНЕННОСТЬ ФУНКЦИОНАЛОМ.

DragonFly основана на FreeBSD 4.x. На данный момент существенное отличие этой ОС от своих собратьев — наличие уникальной модели легковесных нитей ядра (LWKT). В такой модели на каждый процессор выделяется независимый планировщик задач, а каждому процессу ставится легковесный поток внутри ядра.

Главным источником информации считается официальное руководство: **leaf.dragonflybsd.org/~justin/handbook/**. Интересную информацию о проекте можно найти на wiki-страничке: wiki.dragonflybsd.org. Хронологию развития ядра можно найти здесь: **wiki.dragonflybsd.org/index.php/User:Jgarcia/Status_Page_Devel**. Довольно подробное описание установки и использования DragonFly содержится в серии статей Алексея Федорчука: **unix.ginras.ru/bsd/dfbsd000.html.**

➔ **TrustedBSD.** В рамках проекта TrustedBSD (**http://trustedbsd.org**) создаются и отлаживаются такие передовые фичи, как UFS2, GEOM, OpenBSM, OpenPAM. Для «настольного» и серверного применения пока не пригоден. В первую очередь, это ось-полигон для разработчиков FreeBSD.

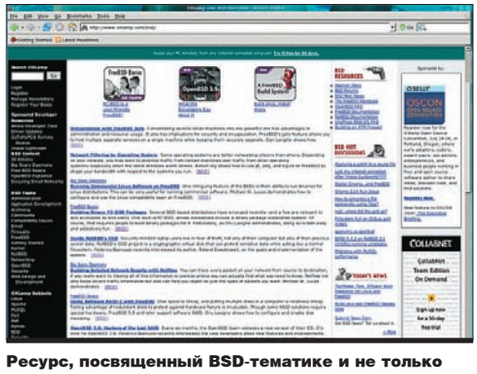
➔ **Frenzy.** Не стоит забывать и про Frenzy (**http://frenzy.org.ua**) — LiveCD на базе ОС FreeBSD. Дистрибутив представляет собой швейцарский нож, своеобразный портативный инструмент хакера или системного администратора с набором программного обеспечения для настройки, проверки и анализа сети, тестирования компьютерного железа.

➔ **PCBSD** — это дистрибутив, ориентированный на использование FreeBSD в качестве десктопной операционки. Другими словами, для самых обычных пользователей и всех тех, кто давно мечтал и боялся познакомиться со славным семейством BSD-систем. Сайт проекта: **http://www.pcbsd.org** .

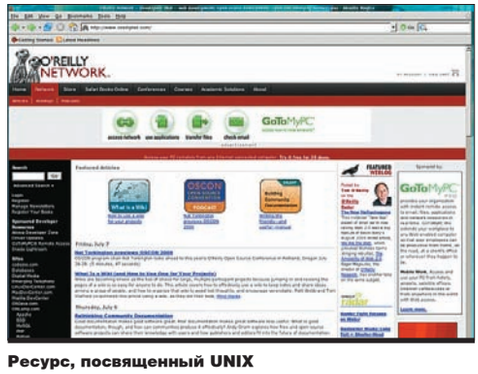
Это далеко не полный список BSD-подобных систем. Как можно заметить, каждый дистрибутив предназначен для определенных нужд и задач. И дело каждого, какую BSD он для себя предпочтет. В интернете существует большое количество информации по BSD-системам. И только ленивый не сможет их найти. Русскоязычными ресурсами, содержащими огромное количество информации и ссылок по BSD-тематике, являются **http://bsdportal.ru** и **http://www.opennet.ru/mp/bsd/**

Из англоязычных рекомендуем посетить следующие ресурсы: **onlamp.com/bsd** и **oreillynet.com**.

Кроме того, на каждом официальном ресурсе существуют списки рассылок (так называемые Mailing Lists), на которые может подписаться любой желающий **C**



Ресурс, посвященный BSD-тематике и не только



Ресурс, посвященный UNIX

муки оптимизации

ОТ СЕРВЕРА К ДЕСКТОПУ

XBSD-СИСТЕМЫ РУЛЯТ НА СЕРВЕРАХ, НО НА ДЕСКТОПАХ ОНИ ОТДЫХАЮТ. САМИ ОЧЕНЬ ТРЕПЕТНО ОТНОСИМСЯ С XBSD (ОСОБЕННО К FREEBSD 4.5), ПОТОМУ ЧТО ЛЮБИМ КОНСОЛЬ И ПРИВЫКЛИ РАБОТАТЬ С КОМАНДНОЙ СТРОКОЙ, КОТОРУЮ НЕ ПРОМЕНЯЕМ НА ФАЙЛОВЫЕ МЕНЕДЖЕРЫ. ОТ BSD В ОСНОВНОМ ТРЕБУЕТСЯ КОМПИЛЯТОР GCC. ДЛЯ РАБОТЫ ЭТОГО ВПОЛНЕ ДОСТАТОЧНО, А ВОТ ДЛЯ РАЗВЛЕЧЕНИЙ — УВЫ!

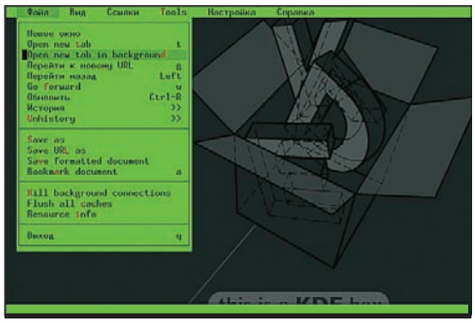
КРИС КАСПЕРСКИ АКА МЫЩЪХ

Типичный пользователь будет чувствовать себя весьма неуютно, если не сможет запустить WinAmp или посмотреть видеофильм. А еще редактировать офисные документы, открывать pdf...

Короче, если использовать xBSD в качестве основной операционной системы, нам обязательно понадобиться GUI, иначе... это уже какой-то извращенный садомазохизм получится, с добровольной самоизоляцией от цивилизованного мира.

➜ **за и против.** Ругая Windows, которая «тормозит» и занимает столько места на диске, сколько находит, мы забываем, что KDE (или GNOME — если кто хочет играть, а не работать) тормозит еще сильнее, а места занимает столько, что...

В отличие от LINUX, в xBSD «десктопное окружение» не ставится из коробки и требует не только танцев с бубном, но еще и толстого канала, поскольку качать придется столько, что стоимость трафика может запросто превысить цену лицензионной копии Windows Professional. Если последняя сразу ставится и работает, то xBSD требует кучи библиотек, поставленных в исходных текстах и влекущих обширные зависимости, которые, в свою очередь, влекут свои зависимости. Причем старые (и легкие) версии библиотек зачастую не удается откомпилировать той версией компилятора, что поставляется в свежих релизах xBSD, а новые — намного тяжелее, да и компилируются сложнее. Даже имея опыт работы с xBSD, собрать из нее десктоп за один день практически нереально. Одна лишь перекачка из сети и компиляция потребуют гораздо большего времени! О том, что каждая откомпилированная программа требует тестирования, скромно промолчим...



Попсовый links, выполненный в псевдографической стилистике

Но эти проблемы меркнут на фоне проблем с оборудованием, и в первую очередь — с видео-картами. Единственная компания, которая их изредка пишет, — это NVIDIA, остальные же просто не обращают на xBSD внимания, поскольку на рынке десктопов она занимает очень узкий сегмент. Без родных или хотя бы реверсированных драйверов не удастся задействовать наивысшее разрешение и аппаратные фичи, ограничившись стандартным VGA или VESA-режимами. Разрешение, впрочем, можно настроить и вручную (если, конечно, знать, какие параметры для этого необходимо передать карте). С частотой развертки дела обстоят чуть сложнее, и приходится либо покупать LCD-монитор (у которого такого понятия просто нет) или подбирать необходимые параметры вручную. Все это требует знаний и времени, а время — деньги.

Ладно, будем считать, что KDE мы все-таки запустили, пускай и не без мата :). Остается найти программы. А с программами дела обстоят довольно туго. Если под LINUX имеется хоть что-то (немного офисных пакетов, 1С бухгалтерия, паратройка 3D-игр), то под BSD нет вообще ничего. Правда, есть возможность запускать LINUX-приложения в режиме совместимости, однако... тормоза при этом резко усиливаются, а некоторые программы вообще не запускаются. В особенности, это касается эмуляторов Windows, без которых даже под LINUX'ом мало что можно сделать.

таким образом, установка xBSD на десктоп:

1. ТРЕБУЕТ ОПЫТА РАБОТЫ С СИСТЕМОЙ;
2. ОТНИМАЕТ КУЧУ ДЕНЕГ И ВРЕМЕНИ;
3. ОГРАНИЧИВАЕТ «КРУГОЗОР» НЕБОЛЬШИМ ПАКЕТОМ ПРОГРАММ.

Зачем же искусственно создавать себе проблемы? Из любви к системе? Так из-под KDE она практически ничем не отличается от того же LINUX'а. Только нормальный дистрибутив LINUX'а ставится сразу.

Сказанное нисколько не умоляет серверные возможности xBSD (серверу KDE нахрен не нужен) или чистой командной строки, которой для работы вполне достаточно. Если из LINUX'а рабочую станцию можно строить из ненависти к Microsoft'y, любви к UNIX-системам или просто от бедности, то никаких убедительных мотивов для превращения xBSD в то, чем она не является, просто нет, да и не будет! Заметь: сами разработчики xBSD не позиционируют свою систему как десктопную. Правда, существует такая штука, как PC-BSD, делающая определенные шаги в десктопном направлении и, по слухам, устанавливающаяся из коробки, но «интеллектуальность» установщика до LINUX'а все-таки не дотягивает. И если LINUX сегодня легко ставят даже те, кто не имеет опыта вообще, то овладеть xBSD с лету не удастся!

В частности, FreeBSD 5.4 по умолчанию устанавливает уровень безопасности в 1 (даже если при установке ей открытым текстом сказать, что секьюрность идет в топку), делающий невозможным запуск X'ов (точнее Xorg) даже из-под root'а. Вываливается невразумительная жалоба на невозможность открытия /dev/io.

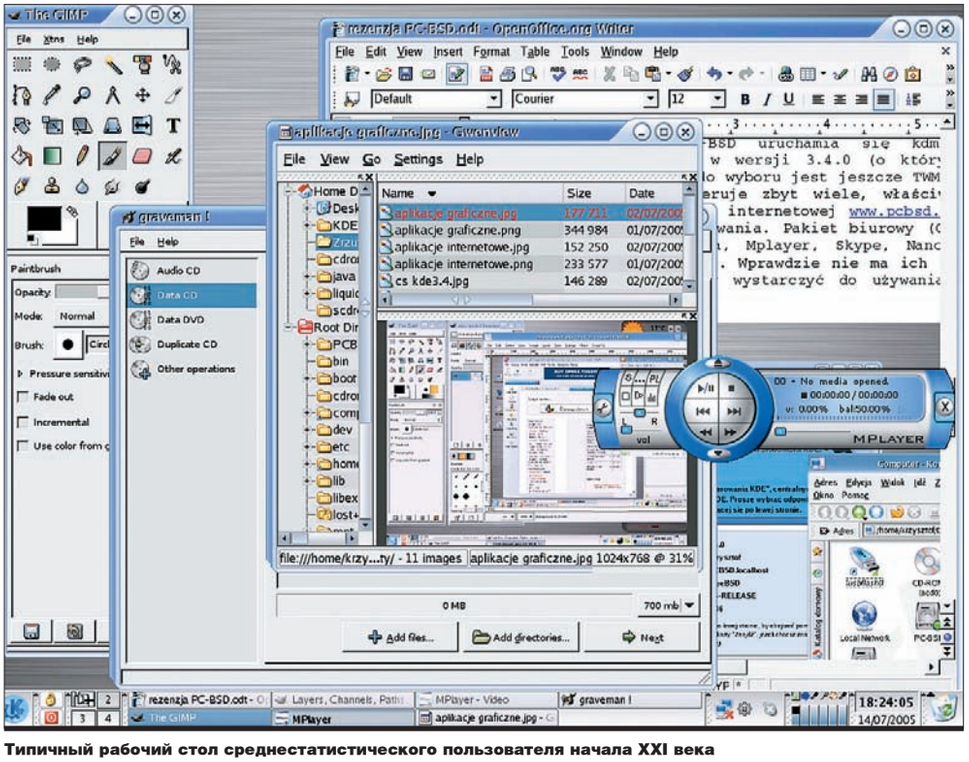
Приходится лезть в /etc/rc.conf и securelevel ручками, а для этого необходимо знать, как устанавливаются и конфигурируются X'ы. И новичок, впервые столкнувшийся с такой проблемой, просто не будет знать, откуда следует рыть. В общем, приятное времяпрепровождение гарантируется :). При этом ничто не мешает держать xBSD (вместе с другими необходимыми для работы системами) на виртуальной машине типа VM Ware, запускаемой откуда угодно — хоть из-под LINUX, хоть из-под Windows!

Другими словами, те, кому xBSD нужна для работы, вполне могут позволить себе раскошелиться на отдельную машину или запускать ее из-под эмулятора. Остальным же рекомендуется либо Windows, либо (при наличии совести и желания приобщиться к миру свободного ПО) LINUX. Кстати, опыт работы с LINUX'ом не очень-то помогает общению с xBSD, поскольку многое в них реализовано неодинаково, в том числе и консоль.

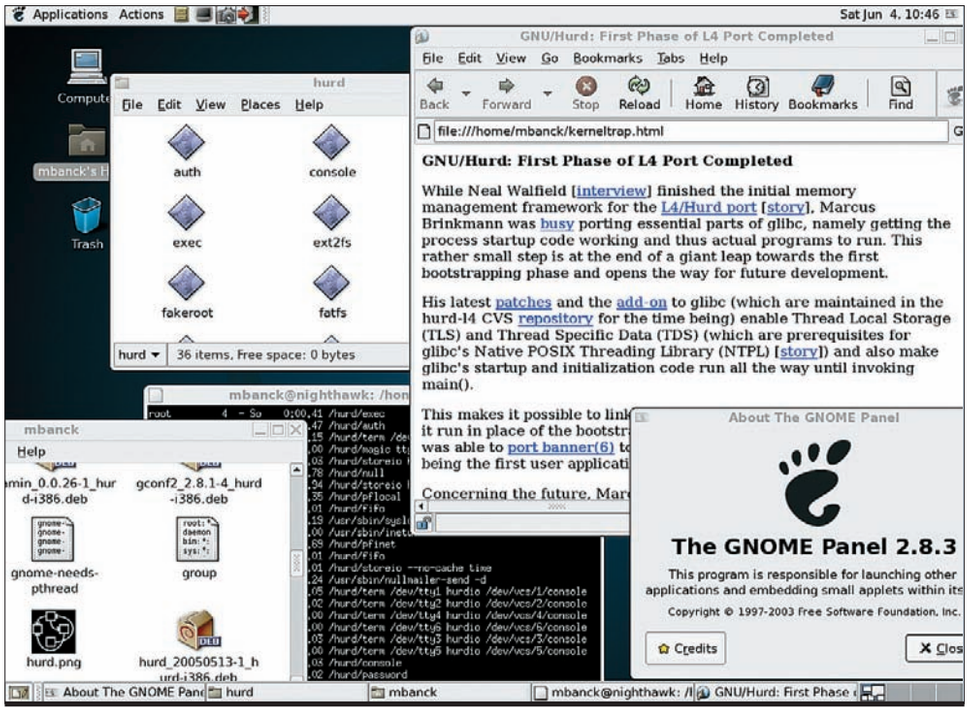
Впрочем, справедливо и обратное. Монтирование дискеты под xBSD осуществляется утилитой mount_msdos (почему-то переименованной в последних версиях в mount_msdosfs), которой в LINUX просто нет, что вызывает удивление и страшно напрягает. Кстати, «типичный пользователь UNIX'а постоянно вспоминает, как называется команда print на этой неделе» (с). Цитате — 25 лет :).

➔ **выбор компонентов.** Главной ошибкой большинства начинающих пользователей xBSD (как, впрочем, и LINUX) является упорное нежелание (или неумение) работать с литературой. Ладно, не хочешь читать объемное руководство по установке, но хотя бы FAQ можно прочесть?! В отличие от LINUX'а, установить xBSD в интерактивном режиме практически невозможно!

Также ни в коем случае не следует впадать в другую крайность — крутить при первой установке xBSD те настройки, которые не до конца понимаешь. Лучше выбрать экспресс-установку и, поработав некоторое время с системой, начинать подгонять ее под себя. То же самое относится и к LINUX'у. Как показывает практика, ручной выбор устанавливаемых пакетов идет только во вред или, в лучшем случае, насмарку, поскольку одни пакеты через зависимости тянут другие, и в результате устанавливается даже то, от чего ты категорически отказался. А то, что хо-



Типичный рабочий стол среднестатистического пользователя начала XXI века



GNOME — оконный менеджер, стремящийся «догнать и перегнать» Windows

тел установить, не работает, потому что инсталлятор не был как следует протестирован и не смог отследить все зависимости, и кое-что осталось недоустановленным.

В частности, начиная с версии 3.0 (более ранних ты все равно не найдешь), компиляция модулей требует наличия исходных текстов ядра, которые простому смертному пользователю вроде бы ни к чему — многие их просто не ставят, а потом дивятся, почему модули (входящие в состав других пакетов) не компилируются.

Лучше всего иметь два диска, повешенные на различные IDE-каналы: один под временные файлы (swap, /var, etc), другой под файлы системы и свои «домашние». Некоторые материнские платы поддерживают три IDE-канала, что позволяет выделить swap в отдельное «делопроизводство», однако, если на компьютере установлено хотя бы 512 Мбайт оперативной памяти, желания посвопить у xBSD практически не возникает, во всяком случае на «домашних» задачах и при компиляции приложений. Кстати, сама xBSD при установке рекомендует выделить под swap пространство, равное удвоенному объему оперативной памяти. Рекомендация странная и совершенно непонятная. Здравый смысл подсказывает, что размер swap-файла, в первую очередь, зависит от максимального объема требующейся виртуальной памяти, которая складывается из размера swap'а и величины ОЗУ. То есть чем меньше у нас оперативной памяти, тем жирнее должен быть swap, но никак не наоборот!

Если виртуальной памяти не хватит, затребовавшее ее приложение просто завершится с ошибкой. Выделять же 512 x 2 == 1024 Мб памяти на подкачку совершенно нецелесообразно и необязательно! Потому что при разбивке по умолчанию swap-раздел располагается между корневым и всеми остальными разделами, а это значит, что головке диска придется совершать перемещения на более далекие дистанции, вызывающие ничем не оправданное снижение производительности.

Если позволяет дисковое пространство, лучше всего выбирать полную установку, а уже потом удалять ненужное.

➔ **разбивка диска.** Дисковая подсистема — узкое место, и разбивка разделов во многом определяет производительность. Если есть возможность, то стоит использовать SCSI-винчестеры, поскольку у них более мощный планировщик запросов, чем в IDE, в результате чего компиляция приложений занимает существенно меньше времени. Если же ты собираешься заниматься частой компиляцией, то оптимальным выбором окажется все-таки IDE с параллельным интерфейсом. SATA-контроллеры все еще достаточно сыроваты и содержат кучу ошибок, приводящих, в том числе, к потерям данных, причем потери могут быть весьма интересными. Так, некоторые контроллеры при определенных обстоятельствах теряют последние несколько байт в последнем секторе переданного блока, в результате чего файл записывается некорректно. Но, если он занимает не весь кластер целиком, некоторое время ошибка остается незамеченной и проявляется только потом. Производители дешевых чипсетов с интегрированным SATA-контроллером (VIA. SiS) предпочитают исправлять такие ошибки в драйве-

SPECIAL ОБЗОР

MEDIUM

LINUX-СЕРВЕР СВОИМИ РУКАМИ

СПб.: Наука и Техника, 2006 / Колисниченко Д.Н.
752 страницы
Разумная цена: 247 р.

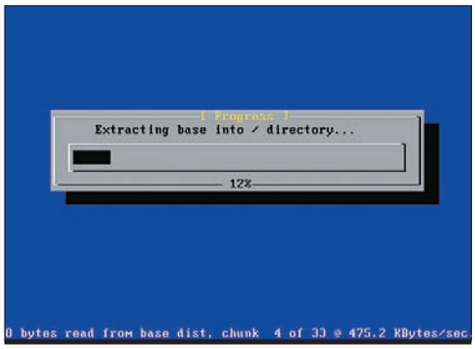
Если возникла необходимость разобраться в настройках Linux-сервера, то имеет смысл делать это последовательно и по данной книге. Те, кто имеет некоторый опыт работы с ОС Linux, могут изучать материал выборочно — кому что нужно. Организация сервера и настройка серверного программного обеспечения. Сетевые настройки сервера, настройка и компилирование серверного ядра. Сервер для Windows-сетей, игровой сервер. Обеспечение безопасности сервера и его служб. Управление трафиком. И все это на основе дистрибутивов Red Hat, Fedora Core и Mandrake.

HARD

QNX/UNIX: АНАТОМИЯ ПАРАЛЛЕЛИЗМА

СПб.: Символ-Плюс, 2006 / Олег Цилюрик
288 страниц
Разумная цена: 273 р.

Адресовано программистам в различных ОС UNIX. Предлагается более детально рассмотреть возможности параллельной организации вычислительного процесса в традиционном программировании. Особый акцент на потоках (threads). На примере реальных кодов показаны приемы и преимущества параллельной организации вычислительного процесса. Для изучения материала не помешают знания языка программирования C/C++ и понимание устройства современных многозадачных ОС UNIX. А в качестве испытательной площадки для тестов выбрана ОС QNX.



Установка FreeBSD в текстовом режиме

рах, естественно, выпущенных только для Windows и, возможно (хоть и маловероятно), для LINUX. Поэтому не бери SATA для xBSD, если полностью не уверен в безглючности.

Раздел /usr лучше всего размещать вплотную к корневому, а /swap, /var и /tmp — держать на отдельном диске. Причем /tmp должен идти после /var, а не наоборот. Мотив /var не требует большого пространства (100 Мб будет более чем достаточно), а вот под временные файлы следует отвести побольше, можно даже весь оставшийся объем. Если расположить /var после /tmp'a, то головка диска будет сильно порхать, поскольку ей придется при каждом обращении /tmp <-> /var пересекать весь /tmp. Разумнее оптимизировать раскладку разделов. То же самое относится и к /swap, который лучше расположить в начале диска, выделив под него сколько-нибудь памяти. «Сколько-нибудь» — потому что очень трудно убедить установщик, что подкачка нам не нужна, и мы предпочитаем все данные хранить в оперативной памяти.

➔ **пересборка ядра.** Все операционные системы семейства xBSD имеют монолитное ядро с опциональной поддержкой модулей, причем большая часть модулей может быть как включена непосредственно в само ядро, так и представлена динамически загружаемыми файлами. По умолчанию GENERIC-ядро включает в себя поддержку практически всего известного ему оборудования, в которое входят и SCSI-контроллеры, и сетевые карты, и другое оборудование, которое встречается только на серверах. Не говоря уже обо всех мыслимых и немыслимых сетевых протоколах, потребность в которых даже на серверах возникает далеко не всегда. Естественно, что это не проходит даром, и за поддержку приходится расплачиваться временем загрузки и потребляемой памятью. С другой стороны, динамические модули грузятся еще дольше. Поэтому наилучшей стратегией будет выбор такой конфигурации ядра, чтобы все часто используемые компоненты включались в него, а редко используемые — выносились в модули.

Управление ядром осуществляется путем прямого редактирования конфигурационных файлов GENETIC и LINT, расположенных в каталоге /sys/i386/conf, с последующей перекомпиляцией. Настроек, прямо или косвенно влияющих на производительность, очень много, поэтому ограничимся наиболее характерными.

Выбор процессора (параметр cpu в разделе CPU Options), равно как и поддержка технологии SSE (параметр CPU_ENABLE_SSE), вопреки слухам, практически ни на что не влияет, поскольку ядро само по себе не использует мультимедийных инструкций, а активация SSE фактически всего лишь указывает на необходимость сохранения SSE-регистров при переключении контекстов, а это уже тормоза. С другой стороны, выключение SSE в ядре делает работу двух и более «мультимедийных» приложений невозможной, и они постоянно гробятся.

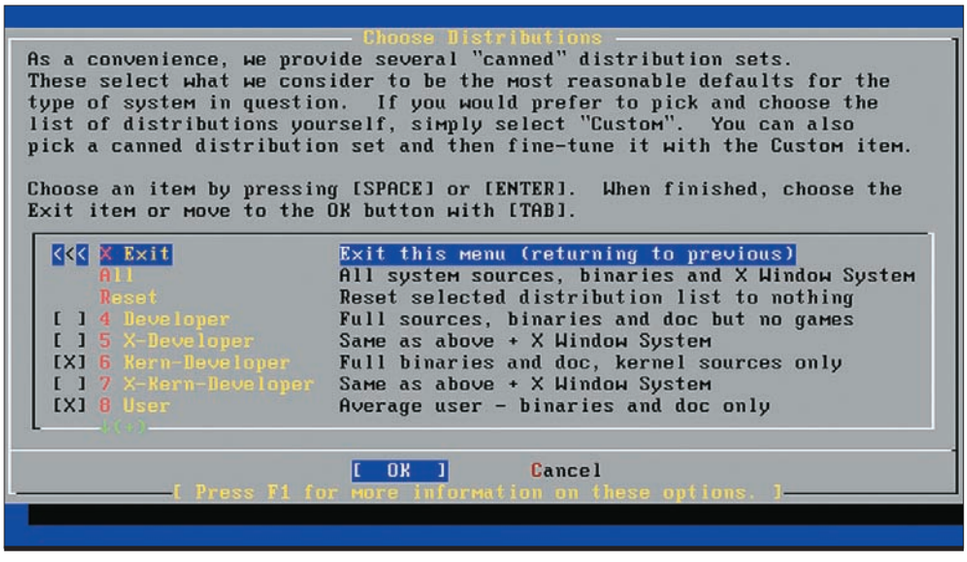
А вот задействовать поддержку многопроцессорных машин (секция SMP Options) на многоядерных кристаллах однозначно стоит! С Hyper-Threading все намного сложение, и наперед очень трудно сказать, принесет ли оно увеличение производительности или нет. На некоторых приложениях наблюдается устойчивое замедление, некоторые не реагирует на это вообще. Так что все решает эксперимент.

Остальные опции относятся к такому типу оборудования, которого по умолчанию поддерживается слишком много, но далеко не все можно безболезненно убирать. В частности, «продергивая» список SCSI-устройств, нельзя забывать, что Zip на параллельном порту тоже относится к SCSI-устройствам (точнее, с помощью драйвера нижнего уровня изображает из себя таковое), поэтому, если отключить поддержку SCSI-устройств (как это делают многие пользователи, имеющие только IDE), ядро не сможет увидеть Zip и... Так и рождаются легенды о том, что xBSD не дружит с Zip'ом, и никаких драйверов для него нет.

Также нельзя отключать поддержку ISA-шины, которая есть даже в тех компьютерах, в которых ее нет. В смысле, на физическом уровне нет (она не распаяна на плате), но куча устройств типа динамика или клавиатуры до сих пор «висят» на ISA-шине, эмулируемой южным мостом чипсета. Так что «оптимизировать» ядро следует с умом, обращая внимание на комментарии в мелочах и предварительно ознакомившись с архитектурой IBM PC в целом.

➔ **оптимизация.** Увлекательное занятие, отнимающее кучу времени на компиляцию и многочисленные эксперименты. Ведь с первого раза собрать оптимально работающее ядро вряд ли получится. Зато потом... можно ускорить систему в несколько раз! А можно ничего не выиграть вообще. Это уж от оптимизатора и мощности оборудования зависит. Тут главное — не перестараться и не потратить на оптимизацию больше времени, чем она в принципе способна отдать назад. Тем более не стоит ковырять стабильно работающую ось, если в этом нет жизненно важной необходимости. Как говорили древние: «Не лови рыбу на золотой крючок». Если он оборвется, никакой улов не компенсирует потери.

Неправильно собранное ядро запросто может перестать загружаться. Задумайся, сможешь ли ты починить систему, не теряя текущих настроек и не прибегая к переустановке. Лучше всего заниматься оптимизацией на только что установленной системе, поскольку в этом случае терять практически нечего. Чрезмерное увлечение оптимизацией всегда приносит больше вреда, чем пользы. Потребность в памяти у BSD довольно невелика, и основное внимание лучше уделить дисковой подсистеме. Более мощный процессор также не помешает, а вот от 64-разрядных камней на рабочих станциях никакого толку, по сути, нет, тем более что 64-разрядный код занимает больше места в памяти, чем 32-разрядный. Добавь сюда проблемы совместимости (64-разрядные порты пока что недостаточно отлажены) и получишь ответ на вопрос, стоит ли переходить на «модную» архитектуру или нет **C**



Устанавливай исходные тексты ядра, даже если не собираешься в них ковыряться. Они необходимы для компиляции модулей

последняя битва

СРАВНЕНИЕ LINUX И BSD ПОД ОСОБЫМ УГЛОМ

ЦЕЛЬ ДАННОЙ СТАТЬИ СОСТОИТ НЕ В ТОМ, ЧТОБЫ ОТВЕТИТЬ НА ВОПРОС: «ЧТО ЖЕ ЛУЧШЕ?», И НЕ В ТОМ, ЧТОБЫ РАЗРЕКЛАМИРОВАТЬ ОДНУ ИЗ СИСТЕМ, А В ТОМ, ЧТОБЫ ПРОВЕСТИ СРАВНЕНИЕ ОСОБЕННОСТЕЙ LINUX И BSD И ПОМОЧЬ ЧИТАТЕЛЮ УВИДЕТЬ МИР UNIX СО ВСЕХ РАКУРСОВ — С ТОЧКИ ЗРЕНИЯ КАК ЛИНУКСОИДА, ТАК И BSD'ШНИКА

ЕВГЕНИЙ ЗОБНИН АКА J1M
{J1M@LIST.RU}

→ **истоки.** Начнем с того, что попробуем определить разницу между понятиями «UNIX-клон» и «UNIX-подобная ОС» и разберемся, какое из понятий к какой из рассматриваемых ОС применимо. Для этого нам придется совершить экскурс в прошлое, как раз в тот момент, когда исходные тексты UNIX попали в калифорнийский университет Беркли. Произошло это знаменательное событие в 1974 году и вскоре началось развитие второй ветки оригинального UNIX, распространялась которая под именем BSD (Berkely Software Distribution). К началу 90-х в BSD-ветке UNIX накопилось столько изменений и улучшений, что было принято решение создать полностью открытую операционную систему, избавившись от кусков кода оригинального UNIX (по условиям лицензии, исходные коды UNIX не могли распространяться дальше университетских стен). Так на свет появилась 386BSD, а затем из нее выросла FreeBSD. Как видно, BSD-системы — это прямые потомки и наследники традиций оригинального UNIX от AT&T, что дает им право именоваться «UNIX-клонами».

История Linux, в отличие от истории BSD, достаточно прямолинейна и даже романтична.

В 1991 году обычный финский студент, фанат программирования Линус Торвальдс, находясь под влиянием учебной ОС minix, бросает миру вызов и в одиночку пишет собственную операционную систему. Вскоре он выкладывает свое творение во всеобщий доступ под открытой лицензией GPL и сообщает об этом группе новостей comp.os.minix. К разработке присоединяется множество людей, и студенческая игрушка постепенно превращается в серьезную ОС. В этой истории легко заметить одну немаловажную деталь: Linux написан, что называется, с нуля. Сам Линус в своих мемуарах говорит, что во время создания ОС у него на руках даже стандарта POSIX не было, не то что исходников UNIX или его потомка — BSD. Linux — это не UNIX, это ОС, исповедующая традиции UNIX-систем, совместимая со стандартом POSIX, но все-таки не UNIX. Linux — это «UNIX-подобная ОС».

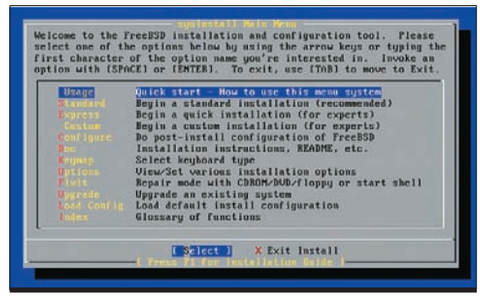
Вдумчивый читатель наверняка заметил ошибку: в последнем абзаце Linux назван операционной системой, хотя каждый должен знать, что Linux — это всего лишь ядро и не более того. В следующем разделе мы попробуем рассмотреть плюсы и минусы распространения ядра Linux как отдельного пакета, и то, чем этот подход отличается от цельных BSD-систем.

→ **интеграция.** С самого начала своего возникновения Linux был всего лишь ядром. Линус Торвальдс не утруждал себя созданием так называемого «окружения ОС», всех тех утилит и библиотек, которые, работая поверх ядра, создают завершенную операционную систему. Вместо этого предлагалось использовать наработки проекта GNU, в рамках которого уже было написано все необходимое, включая стандартную библиотеку языка Си (libc). Так как и ядро Linux, и все программы проекта GNU выпускались под лицензией GPL, то никаких проблем в создании на их основе операционной системы, пригодной для использования, не возникало. Так появилось понятие дистрибутива Linux, то есть окружения ОС на основе ядра Linux. В современные дистрибутивы Linux входит софт не только проекта GNU, но и множество других программ от сторонних разработчиков, лицензии которых совместимы с GPL.

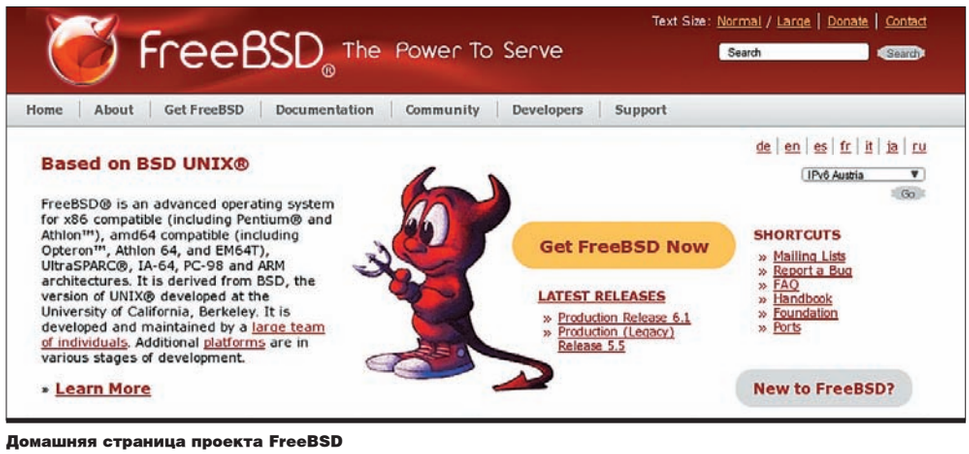
BSD-системы, будучи прямыми потомками UNIX, изначально комплектовались всем необходимым для комфортной работы в ОС. Это подтверждается наличием слова Distribution в самой аббревиатуре BSD. Все системные компоненты окружения ОС разрабатываются одной командой разработчиков, а их развитие координируется лидерами проекта. За счет этого поддерживается общая целостность и непротиворечивость всей операционной системы. Системные утилиты имеют четкие, понятные имена и единообразные флаги. Яркие примеры — vidcontrol и kbdcontrol. По названию утилит легко определить, что они предназначены для настройки консоли и клавиатуры. Никаких дополнительных программ не существует, поведение консоли настраивается только двумя утилитами, начиная от типа курсора и заканчивая цветом и шрифтами.

Взглянем теперь на ситуацию в мире Linux. Системные программы, которыми комплектуется дистрибутив, разрабатываются зачастую совершенно независимыми командами. Представления разработчиков об имени команды и способах управления ею существенно различаются. В результате возникает мешанина — огромное количество утилит с невнятными названиями и различным поведением. Ситуация усугубляется еще и тем, что независимые разработчики не могут скоординировать свою работу, и, как результат, появляется множество различных утилит, управляющих, по сути, одним и тем же устройством. Рассмотрим тот же пример с консолью. Для полной настройки и русификации консоли в Linux придется прибегнуть к помощи аж четырех утилит: loadkeys для настройки раскладки, setfont для смены шрифта, setterm для управления визуальными параметрами и fbset для настройки разрешения графической консоли. Причем если возникнет необходимость сменить разрешение текстовое, то это можно сделать только путем передачи параметра ядру. Хотя, затрагивая вопрос графики в консоли, надо отдать пингвину должное: настоящей графической консоли в FreeBSD нет, максимум, что можно сделать, — это установить VESA-режим с разрешением 800x600 и частотой смены кадров 60Гц. Но ситуация меняется: ребята из проекта DragonFlyBSD доработали консольный драйвер и научили его действовать во всех режимах, которые только позволяет использовать железо машины. Не так давно был сделан бэкпорт этого драйвера в FreeBSD.

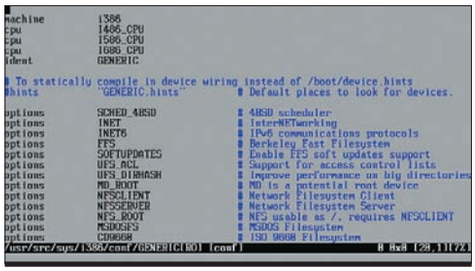
С точки зрения пользователя, BSD-системы — это цельные, хорошо спроектированные и укомплектованные операционные системы. С другой стороны, ОС на базе ядра Linux — это нечто вроде конструктора, детали которого плохо стыку-



Инсталлятор FreeBSD



Домашняя страница проекта FreeBSD



Конфигурирование ядра FreeBSD

ются между собой и не подходят по цвету. Задачу сборки такого конструктора решают дистрибьюторы, а то, к чему это приводит, мы рассмотрим в следующем разделе.

➔ **свобода выбора.** До недавнего времени в мире BSD вообще не существовало понятия «дистрибутив» с тем смыслом, какой в него вкладывают линуксоиды. Пользователю предлагалось на выбор четыре варианта ОС: FreeBSD, NetBSD, OpenBSD и DragonFlyBSD. Каждая из перечисленных ОС была рассчитана на решение определенного круга задач, за исключением FreeBSD, которая позиционировалась как многоцелевая. Ситуация осталась прежней, но на сцене появилось два новых проекта: DesktopBSD и PC-BSD. И это уже не отпрыски семейства BSD, а самые настоящие дистрибутивы FreeBSD с некоторыми улучшениями в плане юзабилити. Их мы рассматривать не будем.

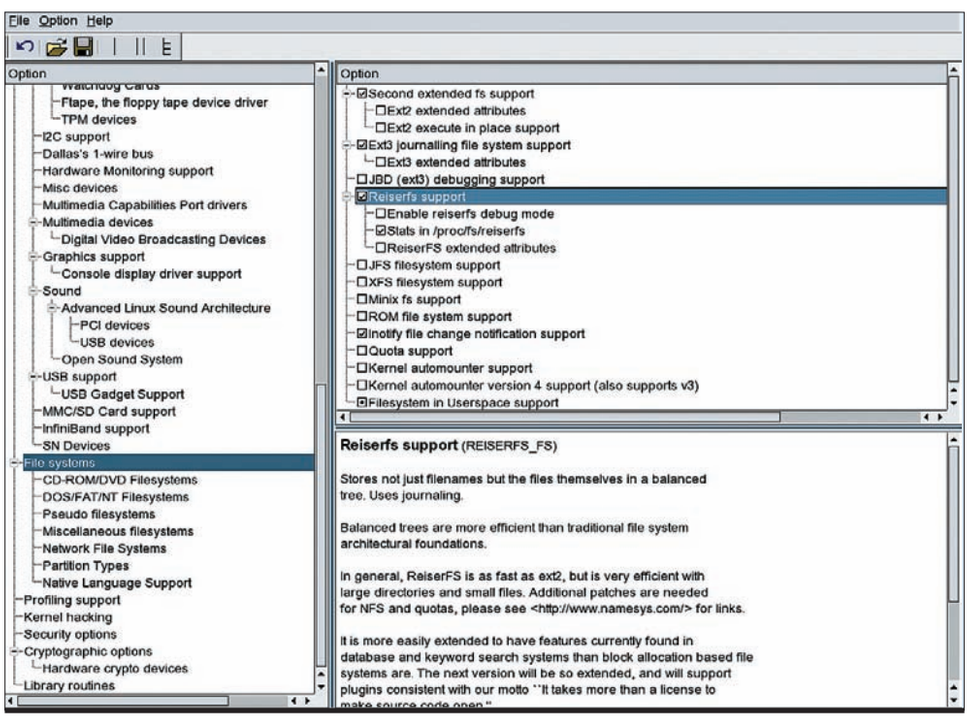
Ситуация с Linux всем известна. Просто колоссальное количество дистрибутивов, каждый со своей историей, целевой аудиторией и, зачастую, собственным форматом пакетов. Каждый дистрибутив Linux может рассматриваться как обособленная UNIX-подобная ОС на базе одного ядра. Перед пользователем открываются огромные просторы для выбора. Каждый может найти для себя тот единственный и неповторимый дистрибутив, который будет удовлетворять всем его потребностям. Одни предпочитают собирать программы из исходников, другим больше по душе прекомпелированные пакеты, для третьих важна простота использования, четвертые предпочитают покопаться во внутренностях пингвина. Выбор практически неограничен.

К сожалению, неограниченная свобода выбора далеко не всегда приносит одну лишь выгоду. Любой пользователь Linux может рассказать множество историй о том, как он, в поисках идеала, прыгал с одного дистрибутива на другой. Сколько времени, нервов и дискового пространства было на это потрачено. Опять же, любой линуксоид знает, что такое поиск нужного пакета для своего дистрибутива. Что такое несовместимость и конфликты между пакетами. И, конечно же, любой опытный линуксоид хоть раз в жизни прибегал к помощи утилиты alien для конвертации пакетов из одного формата в другой. С этим можно не согласиться, сказав, что проблемы надуманы, и любой мало-мальски популярный дистрибутив содержит в базе пакетов все, что только может потребоваться пользователю. Отчасти это правда, но полностью отрицать существование проблемы нельзя.

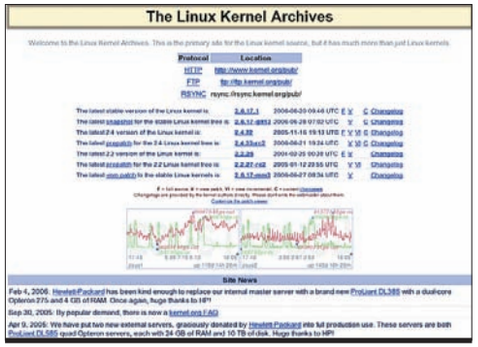
В мире BSD, напротив, выбор очень и очень ограничен. Несмотря на то, что различные представители BSD-семейства разительно отличаются друг от друга в плане архитектуры ядра и целевой аудитории, юзерлэнд-окружение в них практически идентично. Те качества, которые отличают дистрибутивы Linux, для BSD-систем являются сходной чертой. В операционных системах семейства BSD можно выделить три общих черты: инициализационные скрипты в стиле BSD, способ установки стороннего ПО через систему портов, простота. Рассмотрим каждую из них подробнее.

➔ **стили инициализации.** Стиль инициализационных скриптов — это первое, что различается в Linux-дистрибутивах и является общим в BSD. Существует два стиля инициализации: SystemV и BSD. SystemV-стиль пришел к нам из оригинального UNIX и основан на понятии runlevel (наиболее часто употребляемый вариант перевода — «уровень запуска»). Скрипты SystemV представляют собой набор файлов-скриптов, каждый из которых ответственен за определенный этап инициализации (это может быть монтирование файловых систем, запуск сервисов и любая другая задача). В зависимости от уровня запуска стартует только заранее определенная часть этих скриптов (например, на уровне 2 скрипты, ответственные за поднятие сети, управления не получают). Привлекательность скриптов SystemV-стиля заключается в их гибкости. Пользователь может самостоятельно контролировать поведение системы на каждом этапе загрузки. Кроме того, SystemV-стиль идеально подходит для организации параллельной загрузки сервисов. Подавляющее большинство дистрибутивов Linux используют такой стиль инициализации.

BSD-стиль, как легко догадаться из названия, изначально использовался только в BSD-системах. В скриптах BSD-стиля нет понятия runlevel, как нет и модульности. После загрузки ядра демон init передает управление скрипту /etc/rc, и тот проводит систему через все этапы инициализации. Достоинство такого подхода — в простоте реализации и управления. Недостаток — в отсутствии гибкости. С другой стороны, BSD-скрипты со временем приобрели черты SystemV-скриптов, что легко заметить по наличию каталога /etc/rc.d. Сегодня файл /etc/rc — это всего лишь обертка, нужная для того, чтобы поочередно передавать управление скриптам из этого каталога, каждый из которых, как и в случае с SystemV, выполняет свою функцию. Это придает некоторую гибкость процессу настройки инициализации.



Конфигурирование ядра Linux



Хранилище Linux-ядер

➔ **система пакетного менеджмента.** Вторая отличительная черта дистрибутива Linux — это система пакетного менеджмента. Практически все существующие способы установки ПО представлены в мире Linux. Это и пакеты, разворачивающиеся прямо в дерево файловой системы (rpm, deb, tgz), и пакеты, устанавливаемые в выделенные каталоги (наподобие «Program Files» из Windows), и BSD-подобные системы портов (вроде портов Gentoo или ArchLinux), и даже модули файловой системы, применяемые в некоторых LiveCD. Опять же, выбор практически неограничен. Причем особой популярностью у дистрибьюторов Linux пользуются пакеты форматов rpm и deb. В особенности первый, который был включен в стандарт LSB (Linux Standard Base). Учитывая тот факт, что пакет rpm имеет, наверное, самый противоречивый и запутанный формат, его популярность вызывает тревогу.

Что касается BSD, то здесь мы видим обратную картину, причем картину, одинаковую во всех BSD-отпрысках. Системы пакетного менеджмента BSD-систем основаны на понятии портов. Система портов представляет собой некий фреймворк, который позволяет скомпилировать и установить любой портированный программный пакет, используя единый интерфейс. Каждый BSD'шник знает, как легко найти и установить нужную программу через систему портов. Для этого потребуется выполнить всего три незамысловатых действия (на примере FreeBSD): находясь в каталоге /usr/ports, набрать команду «make search name=имя», перейти через дерево портов в нужный каталог и выполнить команду «make install». Это все: никаких команд с непонятными флагами и километровыми man-страницами, никаких поисков пакетов в интернете — все просто и ясно. Более того, не возбраняется и установка перекомпилированных пакетов. Для выполнения этой операции предусмотрена команда pkg_add, которая, будучи запущенной с опцией '-r', вовсе освобождает пользователя от лишних телодвижений, автоматически выкачивая сам пакет и все его зависимости из сети.

Прошли времена, когда система пакетного менеджмента BSD превосходила все существующие наработки Linux-сообщества. Сегодня BSD-подобную систему портов можно встретить во многих Linux-дистрибутивах (Gentoo, ArchLinux, CRUX). Особого внимания заслуживает система портов Gentoo, которая даже превосходит свой прообраз из BSD-систем. Идея apt-get как средства быстрой и простой установки пакетов сегодня применяется во всех популярных дистрибутивах. Одна из сильнейших сторон BSD постепенно уходит на задний план.

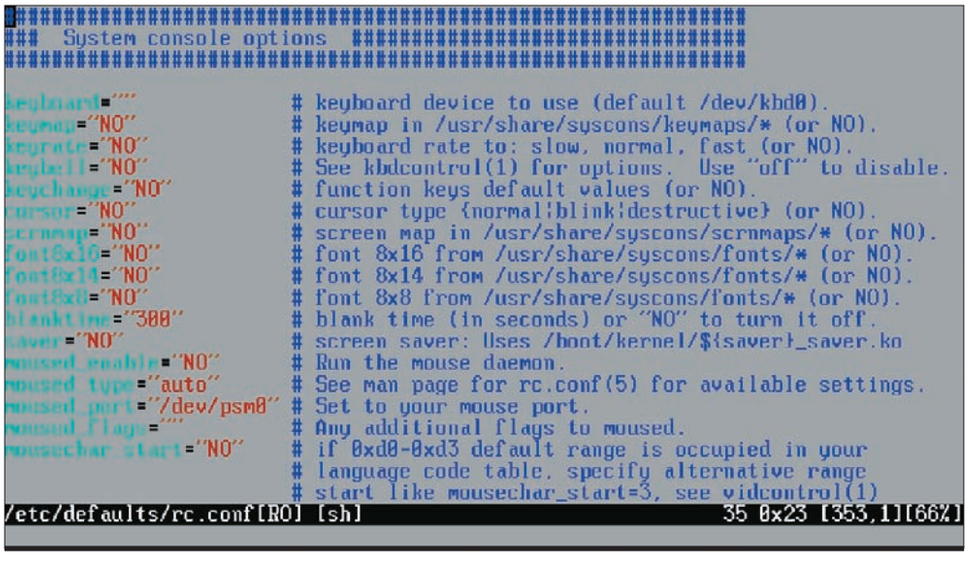
➔ **простота.** И, наконец, последнее, что отличает дистрибутивы Linux и является общим для BSD-систем — простота устройства всей операционной системы. Известна тенденция пользователей (особенно неопытных) делить все существующие дистрибутивы на две группы: простые в использовании, для новичков (SuSe, Mandriva, Ubuntu) и дистрибутивы для профи (Slackware, CRUX, Gentoo). Это тенденция не отражает реального положения вещей, но для нас является значимой. В частности потому, что «дистрибутивы для новичков» обычно очень сложны в архитектурном плане, потому как сама ОС должна решать сложные задачи за пользователя и предоставлять ему различные графические конфигураторы. Дистрибутивы «для профи», наоборот, весьма просты в отношении внутреннего устройства. Отсюда можно сделать вывод: Linux настолько сложен в использовании, что возникла потребность в создании специальных дистрибутивов для людей. Но вывод этот не совсем правильный: дистрибутивы для людей нужны также потому, что нужно скрыть от глаз пользователя ту конструкцию, которая состоит из плохо стыкующихся между собой деталей разного цвета.

BSD-системы, изначально спроектированные как цельные операционные системы, лишены недостатков Linux. Конструктор легко складывается в простую, как в пользовательском плане, так и в плане архитектуры, ОС. Все компоненты ОС — на своем месте и четко выполняют свои обязанности. Можно привести множество примеров, подтверждающих этот факт. Это и простота конфигурирования большинства параметров ОС путем правки файла /etc/rc.conf, и куда более простой механизм загрузки модулей и внятный механизм конфигурирования ядра — sysctl (в противовес запутанной каталоговой структуре /proc/sys). Отдельно стоит упомянуть также о том, что разработчики Linux зачастую слишком усложняют простые вещи.

➔ **документация.** Последний вопрос на сегодня — документированность ОС. Учитывая разительное отличие качества документации Linux и BSD, считаю этот фактор немаловажным. Ситуация с документацией Linux весьма плачевна. Многие нововведения ядра не документируются вовсе, те, для которых документация существует, не обновляются или обновляются нерегулярно. Зачастую бывает очень проблематично найти сведения о нужном драйвере. Любой, кто заглядывал в каталог /usr/src/linux/Documentation, знает, какой там творится беспорядок и как трудно найти нужную информацию. К сожалению, многим утилитам пользовательского уровня свойственна та же проблема. Конечно, нельзя забывать и о документации, распространяемой Linux-дистрибьюторами, но она охватывает только небольшую часть вопросов. Документация проекта **www.tldp.org** (The Linux Documentation Project) — вообще не документация, а руководство «Как сделать то-то». Можно сказать, что Linux берет не качеством, а количеством документации.

Что касается BSD, то здесь все иначе. Особенности каждого драйвера, файловой системы или другой части ядра, с которым приходится непосредственно работать пользователю, доходчиво расписаны в оперативно обновляемых man-страницах. Для FreeBSD существует также регулярно обновляемый handbook (настольная книга, руководство пользователя), который охватывает пусть не все, но очень широкий круг вопросов. В большинстве случаев, чтобы овладеть FreeBSD достаточно только handbook'а и man-страниц. Дополнительные материалы понадобятся только в узкоспециализированных случаях.

➔ **резюме.** Не нужно писать мне гневных отзывов со словами: «Где сравнение ядер?», «Где бенчмарки?», «Где сравнение ФС?» и т.п. Если бы пользователи выбирали ОС только по критериям быстродействия или поддержки новых технологий, то выбор всегда бы падал на Linux. Почему тогда так много людей выбирает BSD-системы? Причину такого выбора и должна была объяснить данная статья **C**



Настройка FreeBSD путем правки файла /etc/rc.conf

оболвань чертенка

ЗАПИСЫВАЕМ CD-R/DVD-R ИЗ FREEBSD

НЕ УСПЕЛ СКАЧАТЬ НОВУЮ ВЕРСИЮ FREEBSD, А У ТВОЕЙ ДВЕРИ УЖЕ ВЫСТРОИЛАСЬ ОЧЕРЕДЬ ЖАЖДУЩИХ С БОЛВАНКАМИ В РУКАХ! ТЫ ЗАПУСКАЕШЬ КЗВ, ВСТАВЛЯЕШЬ ПЕРВЫЙ ДИСК, ПРОХОДИШЬ МЫШКОЙ ПО КНОПКАМ — И ПЕРВЫЙ ДИСК ГОТОВ, ВСТАВЛЯЕШЬ ВТОРОЙ, ОПЯТЬ НАЖИМАЕШЬ КНОПКИ — И РЕЖЕТСЯ ВТОРОЙ ДИСК, ПОТОМ ДЕСЯТЫЙ. КАК ВСЕ ЭТО НАДОЕЛО — КРИЧИШЬ И ВЫГОНЯЕШЬ ВСЕХ ПРОЧЬ. НО ЗАЧЕМ ТАК ТРАВМИРОВАТЬ ПСИХИКУ, ЕСЛИ ТЕХ ЖЕ РЕЗУЛЬТАТОВ МОЖНО ДОБИТЬСЯ, НАБРАВ ВСЕГО ОДНУ КОМАНДУ В ТЕРМИНАЛЕ?

ЕВГЕНИЙ ЗОБНИН АКА J1M
{J1M@LIST.RU}

При использовании утилит командной строки для прожига дисков нужно учесть несколько особенностей этого дела. Во-первых, перед записью диска необходимо создать его образ, то есть файловую систему, которая будет помещена на диск. Для этого мы будем использовать программу mkisofs. Во-вторых, существует несколько утилит, предназначенных для непосредственной записи образа на диск. В статье мы рассмотрим штатную программу burncd, а также cdrecord, которую можно установить из портов. В-третьих, burncd и cdrecord не подходят для записи DVD-дисков. Их мы будем резать при помощи growisofs из пакета dvd+rw-tools.

➔ **создаем iso-образ.** Утилита mkisofs, нужная нам для создания файловой системы диска, находится в пакете cdrtools вместе с программой cdrecord. Поэтому обратимся к системе портов:

```
# cd /usr/ports/sysutils/cdrtools
# make install -DMKISOFS
```

Утилита mkisofs довольно проста в использовании, но принимает очень много флагов. Самые важные из них можно увидеть на соответствующей врезке.

Последние три опции совсем не обязательны. Обычно они используются при создании дисков с ОС на борту. Например, для двухдисковой версии FreeBSD, в поле Application ID может значиться «FreeBSD 5.5», а Volume ID будет различно: «Disk 1» и «Disk 2». Для нас эти опции не представляют интереса, мы будем использовать только флаги -J (для совместимости с Windows) и -R (для совместимости с UNIX).

Для создания образа достаточно выполнить такую команду:

```
$ mkisofs -J -R -o cd.iso каталог
```

В результате файл cd.iso будет содержать все файлы указанного каталога. Обязательно убедись, что размер образа не превышает 700 Мб.

➔ **прожигаем.** Образ диска и пустой компакт-диск готовы, пора приступать к прожигу болванки. Сперва разберем пример со стандартной программой burncd. Она поддерживает несколько флагов и команд.

Рассмотрим их подробнее:

-f — позволяет указать файл устройства резака;
-m — создание мультисессионного диска;
-s — скорость резака (стандартное значение — 1, чтобы включить максимальную скорость, необходимо указать max);
-t — тестовый режим.

При записи также необходимо использовать несколько команд:

data — указывает на то, что мы хотим записать обычные данные;
audio — пишем аудиотреки;
fixate — генерация записи TOC (Table Of Contents), указывается последней.

Запись TOC нужна практически всегда, без нее компьютерный привод не сможет прочитать диск с данными. А в случае создания аудиодиска она не нужна.

Для того чтобы записать на диск наш ISO-образ, применим такую команду:

```
# burncd -s max data cd.iso fixate
```

Не возбраняется также смешивание данных с аудиотреками:

```
# burncd -s max data cd.iso audio
track1 track2 fixate
```

В случае необходимости от промежуточного ISO-образа можно избавиться, пропустив вывод команды mkisofs на вход burncd через канал:

```
# mkisofs -J -R каталог | burncd -s max
data – fixate
```

опции mkisofs

-o — УКАЗАТЬ ИМЯ ISO-ОБРАЗА.

-b — СОЗДАТЬ ЗАГРУЖАЕМЫЙ ДИСК (EL TORITO). СЛЕДОМ ЗА ОПЦИЕЙ НЕОБХОДИМО УКАЗАТЬ ПУТЬ ДО ЗАГРУЗЧИКА, РАЗМЕР КОТОРОГО ДОЛЖЕН БЫТЬ РАВЕН 1200, 1440 ИЛИ 2880 КБ.

-f — СЛЕДОВАТЬ СИМВОЛИЧЕСКИМ ССЫЛКАМ (ЧТОБЫ НЕ ЗАПИСАТЬ НА ДИСК ССЫЛКУ, УКАЗЫВАЮЩУЮ НА ФАЙЛ, РАСПОЛОЖЕННЫЙ НА ЖЕСТКОМ ДИСКЕ).

-J — РАСШИРЕНИЕ JOLIET ОТ MICROSOFT. ИМЕНА ФАЙЛОВ В КОДИРОВКЕ UNICODE, ДЛИНА ИМЕН — 64 СИМВОЛА.

-R — РАСШИРЕНИЕ ROCK RIDGE. ВОЗМОЖНОСТЬ СОЗДАТЬ НА ДИСКЕ ФАЙЛЫ УСТРОЙСТВ, СИМВОЛИЧЕСКИЕ ССЫЛКИ, А ТАКЖЕ НАЗНАЧИТЬ ФАЙЛАМ ПРАВА ДОСТУПА.

-r — ТО ЖЕ, ЧТО И -R, НО С НЕКОТОРЫМИ ИСКЛЮЧЕНИЯМИ. ВЛАДЕЛЕЦ ФАЙЛА И ГРУППА ВЫСТАВЛЯЮТСЯ В НОЛЬ, ПРАВА НА ЧТЕНИЕ ВЫСТАВЛЯЮТСЯ ДЛЯ ВСЕХ ФАЙЛОВ, ВСЕ БИТЫ ЗАПИСИ И СПЕЦИАЛЬНЫЕ БИТЫ ОЧИЩАЮТСЯ. ЕСЛИ В ПРАВАХ НА ФАЙЛ ХОТЬ ОДИН БИТ ИСПОЛНЕНИЯ УСТАНОВЛЕН, УСТАНАВЛИВАЮТСЯ И ДРУГИЕ. ВСЕ ЭТО НУЖНО ДЛЯ ТОГО, ЧТОБЫ ДИСК МОЖНО БЫЛО ПРОЧИТАТЬ В ЛЮБОЙ UNIX-СИСТЕМЕ, НЕЗАВИСИМО ОТ ИЗНАЧАЛЬНЫХ ПРАВ ДОСТУПА.

-A — APPLICATION ID, НАЗВАНИЕ ЗАПИСЫВАЕМОЙ ПРОГРАММЫ.

-V — VOLUME ID, МЕТКА ТОМА.

-p — PREPARER ID, ИМЯ СОЗДАТЕЛЯ ДИСКА.

```
# burncd -v -s max data cd.iso fixate
adding type 0x08 file cd.iso size 1258 KB 629 blocks
next writeable LBA 0
addr = 0 size = 1288192 blocks = 629
writing from file cd.iso size 1258 KB
written this track 1258 KB (100%) total 1258 KB
fixating CD, please wait..
```

Используем burncd для прожига дисков

Но здесь нужно быть осторожным: если данные в канал будут поступать с недостаточной скоростью, то ты рискуешь испортить болванку.

Теперь рассмотрим пример с программой cdrecord. У нее более гибкое управление, но она способна работать только со SCSI-устройствами. Это не проблема: благодаря atapicam в FreeBSD можно выдать любой ATAPI-резак за SCSI-устройство. Кроме того, особенность cdrecord в том, что она адресует резак не через файл устройства, как burncd, а через SCSI-адрес (scsibus,target,lun). Пусть тебя это не пугает, так как, используя команду «cdrecord -scanbus», легко узнать SCSI-адрес резака.

Для начала следует выяснить, присутствует ли в ядре поддержка atapicam. Для этого набираем команду «camcontrol devlist». Если в списке устройств не будет ATAPI-привода — значит, придется пересобрать ядро, добавив в его конфиг строку «device atapicam». Далее набираем «cdrecord –scanbus» и смотрим адрес SCSI-устройства. Его мы и будем использовать в дальнейшем.

Чтобы определить характеристики резака, можно дать команду «cdrecord –checkdrive». Нас интересует полезнейшая технология защиты от опустошения буфера (burnfree). И когда все готово, можно приступать к прожигу диска:

```
# cdrecord dev=1,1,0 speed=52
driveropts=burnfree cd.iso
```

Опция 'speed=52' позволяет разогнать резак на максимально допустимой скорости, «driveropts=burnfree» задействует одноименную технологию.

На этом мы заканчиваем разговор о CD-R и переходим к DVD-R.

Для создания DVD-дисков следует использовать утилиту growisofs из пакета dvd+rw-tools (/usr/ports/sysutils/dvd+rw-tools). Это единственная программа из нашего обзора, работать с которой действительно просто. Чтобы поместить на DVD-диск содержимое каталога, достаточно выполнить одну простую команду:

```
# growisofs -Z /dev/acd0 -R -J каталог
```

Программа сама запустит mkisofs (с флагами -R и -J) для создания ISO-образа и пометит его на диск. Добавить новую сессию не сложнее:

```
# growisofs -M /dev/acd0 -R -J каталог
```

Записать готовый ISO-образ еще проще:

```
# growisofs -M /dev/acd0=cd.iso
```

➔ **мультисессия.** Если с мультисессионными DVD-дисками все очень и очень просто, то с CD-R придется позаморачиваться. Во-первых, возможность добавления новой сессии нужно предусмотреть, не закрывая предыдущую сессию. Для этого следует использовать флаг '-multi' в случае использования cdrecord или '-m' в случае с burncd. Во-вторых, при создании образа для новой сессии команде mkisofs необходимо передать номера начального и конечного секторов, полученные, в свою очередь, при помощи команды «cdrecord dev=1,1,0 –msinfo» или «burncd msinfo». Вся последовательность команд выглядит примерно следующим образом (на примере cdrecord):

[1] ПИШЕМ ПЕРВУЮ СЕССИЮ: «CDRECORD DEV=1,1,0 SPEED=52 DRIVEROPTS=BURNFREE -MULTI CD.ISO»;

[2] СНИМАЕМ ЗНАЧЕНИЕ MSINFO: «MSINFO=`CDRECORD DEV=1,1,0 -MSINFO 2>/DEV/NULL`"»;

[3] СОЗДАЕМ ISO-ОБРАЗ СО ВТОРОЙ СЕССИЕЙ: «MKISOFS -J -R -C $MSINFO -M 1,1,0 -O CD2.ISO»;

[4] ПИШЕМ ВТОРУЮ СЕССИЮ: «CDRECORD DEV=1,1,0 SPEED=52 DRIVEROPTS=BURNFREE -MULTI CD2.ISO» C

```
# cdrecord dev=1,1,0 -checkdrive
Cdrecord-Clone 2.01 (i386-unknown-freebsd5.4) Copyright (C) 1995-2004 JWrg Schilling
scsidev: '1,1,0'
scsibus: 1 target: 1 lun: 0
Using libscg version 'schily-0.8'.
Device type    : Removable CD-ROM
Version        : 0
Response Format: 2
Capabilities   :
Vendor_info    : 'SONY    '
Identifikation : 'DVD RW DW-Q28A  '
Revision       : 'KYS1'
Device seems to be: Generic mmc2 DVD-R/DVD-RW.
cdrecord: This version of cdrecord does not include DVD-R/DVD-RW support code.
cdrecord: If you need DVD-R/DVD-RW support, ask the Author for cdrecord-ProDVD.
cdrecord: Free test versions and free keys for personal use are at ftp://ftp.berlios.de/pub
/cdrecord/ProDVD/
Using generic SCSI-3/mmc   CD-R/CD-RW driver (mmc_cdr).
Driver flags   : MMC-3 SWABAUDIO BURNFREE FORCESPEED
Supported modes: TAO PACKET SAO SAO/R96P SAO/R96R RAW/R16 RAW/R96P RAW/R96R
#
```

Характеристики привода

АДСКАЯ СМЕСЬ

в разделе:

ядро — на вилы

ПЕРЕХВАТ СИСТЕМНЫХ ВЫЗОВОВ

ТЕХНИКА ВНЕДРЕНИЯ В ЯДРО BSD И ПРИНЦИПЫ ПЕРЕХВАТА СИСТЕМНЫХ ФУНКЦИЙ МАЛО ЧЕМ ОТЛИЧАЮТСЯ ОТ LINUX И NT, ОДНАКО ЗДЕСЬ ЕСТЬ СВОЯ СПЕЦИФИКА. РАЗБЕРЕМСЯ НА КОНКРЕТНЫХ ЛИСТИНГАХ, УЧИТЫВАЮЩИХ АРХИТЕКТУРНЫЕ ОСОБЕННОСТИ FREEBSD, NETBSD И OPENBSD

КРИС КАСПЕРСКИ АКА МЫЩЪХ

Ядро изолировано от адресного пространства прикладных приложений, и для взаимодействия с ним операционная система представляет ряд интерфейсов. FreeeBSD и NetBSD имеют монолитное ядро, поддерживающее загрузку динамических модулей, очень похожих на модули LINUX и чем-то напоминающие NT-драйвера. Загрузка модуля осуществляется на лету, не требуя перезагрузки операционной системы, что очень хорошо. Естественно, для этого требуется права root'а, которые необходимо каким-то образом заполучить (но это уже тема отдельного разговора). В GENERIC-ядре OpenBSD модули по умолчанию включены, но многие администраторы собирают монолитное ядро без поддержки модульности, считая, что это увеличивает защищенность, лишая атакующего возможности внедрять в ядро вредоносный код, однако...

Все xBSD-системы поддерживают псевдоустройство /dev/[k]mem (аналогичное тому, что имеет-

```
          response from the standard input begins with the character 'y' or
          'Y', the file copy is attempted.  (The -i option overrides any pre-
          vious -f options.)

# cp -R /usr/share/examples/kld/* /root/A
# man -k KLD
kld(4) - dynamic kernel linker facility
kldconfig(8)            - display or modify the kernel module search path
kldfind(2)              - returns the fileid of a kld file
kldfirstmod(2)          - return first module id from the kld file specified
kldload(2)              - load KLD files into the kernel
kldload(8)              - load a file into the kernel
kldnext(2)              - return the fileid of the next kld file
kldstat(2)              - get status of kld file
kldstat(8)              - display status of dynamic kernel linker
kldsym(2)               - look up address by symbol name in a KLD
kldunload(2)            - unload kld files
kldunload(8)            - unload a file from the kernel
# ls -l /usr/share/examples/kld
total 10
-rw-r--r--  1 root  wheel  3650 Jan 28  2002 Makefile
drwxr-xr-x  4 root  wheel   512 Sep 14  2004 cdev
drwxr-xr-x  3 root  wheel   512 Sep 14  2004 dyn_sysctl
drwxr-xr-x  4 root  wheel   512 Sep 14  2004 syscall
#
```

Запрос «man -k KLD» обнаруживает множество документов, относящихся к загружаемым модулям ядра в FreeBSD

ся в LINIX) и библиотеку функций libkvm для работы с ним, прямых аналогов которой в LINUX нет. Поэтому, даже когда модули недоступны, у нас по-прежнему остается возможность модификации ядра, осуществляемая непосредственно с прикладного уровня, при условии, что мы владеем правами root'а.

➔ **модули.** NetBSD и OpenBSD поддерживают LKM-модули (Loadable Kernel Module), «слизанные» с SunOS 4 и совместимые на интерфейсном уровне, реализованном через псевдоустройство /dev/lkm, с которым прикладные приложения взаимодействуют посредством вызовов ioctl (подробнее в «man 4 lkm»). В OpenBSD модули могут быть загружены только на нулевом уровне безопасности. Если же уровень отличен от нуля, а загрузить модули все-таки необходимо, то следует отредактировать файл /etc/rc.securelevel, загружая модули до того, как уровень безопасно будет установлен на необходимую величину. В этом случае о динамической загрузке следует забыть, расставшись с одним из наиболее элегантных свойств оси.

Модули бывают разных типов:

— SYSTEM CALL MODULES — РЕАЛИЗУЮЩИЕ НОВЫЕ СИСТЕМНЫЕ ВЫЗОВЫ (ИЛИ ЗАМЕЩАЮЩИЕ УЖЕ СУЩЕСТВУЮЩИЕ).

— VIRTUAL FILE SYSTEM MODULES — ПОДДЕРЖИВАЮЩИЕ ВИРТУАЛЬНЫЕ ФАЙЛОВЫЕ СИСТЕМЫ.

— DEVICE DRIVE MODULES — УПРАВЛЯЮЩИЕ СУЩЕСТВУЮЩИМИ ИЛИ НЕСУЩЕСТВУЮЩИМИ ШИНАМИ И УСТРОЙСТВАМИ.

— EXECUTION INTERPRETER MODULES — ОТВЕЧАЮЩИЕ ЗА ЗАГРУЗКУ РАЗЛИЧНЫХ ИСПОЛНЯЕМЫХ ФОРМАТОВ.

— MISCELLANEOUS MODULES — К НИМ ОТНОСЯТСЯ ВСЕ МОДУЛИ, НЕ ПОПАДАЮЩИЕ НИ ПОД ОДНУ ИЗ КЛАССИФИКАЦИЙ. ПОДРОБНЕЕ — В «MAN 9 MODULE».

Каждый тип модуля имеет свои особенности реализации, но нам они без разницы. Управлять оборудованием мы не собираемся, устанавливать новую файловую систему — тоже. Перехват системных вызовов может быть осуществлен из любого модуля, а не только из MOD_SYSCALL. Это можно сделать непосредственно в процедуре начальной инициализации модуля, что избавит от необходимости заполнять все служебные структуры, которые в случае Device Drive модулей довольно громоздки.

Примеры готовых модулей можно найти непосредственно в самой NetBSD/OpenBSD, обратившись к каталогу /usr/share/lkm/, или скачать их напрямую из сети (**www.openbsd.org/cgi-bin/cvsweb/src/share/lkm**). Забавно, но в OpenBSD эти файлы не модифицировались свыше 6 лет! Примеры из NetBSD посвежее будут — «всего» 5 лет выдержки, но, по большому счету, никакой разницы между ними нет, и они практически один в один повторяют друг друга.

➔ **MOD_SYSCALL-модуль.** Рассмотрим скелет простейшего MOD_SYSCALL-модуля, перехватывающего системный вызов #1 (mkdir) и устанавливающего на него свою хакерскую «заглушку», выводящую на экран «rock you» и мерзко пищащую спикером. При желании из нее можно вызвать оригинальную функцию mkdir, передав управление по адресу, сохраненному в переменной old_mkdir.

простейший LMK-модуль, демонстрирующий технику перехвата системных вызовов под NetBSD и OpenBSD

```
/* модуль, перехватывающий mkdir
и работающий под Net- и OpenBSD      */
/* ================================= */
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/ioctl.h>
#include <sys/cdefs.h>
#include <sys/conf.h>
#include <sys/mount.h>
#include <sys/exec.h>
#include <sys/lkm.h>
#include <sys/proc.h>
#include <sys/syscallargs.h>
#include <sys/syscall.h>

/* объявляем переменную old_mkdir,
в которую позже будет записан        */
/* оригинальный адрес системного
вызова mkdir                         */
int (*old_mkdir) (struct proc * p,
void * v, register_t retval);

/* функция-заглушка, устанавливаемая
на место mkdir                       */
int hack(struct proc *p, void *v, int
*retval)
{
  printf ("rock you!\x7\n"); return 0;
}

/* процедура начальной загрузки модуля */
static int load(struct lkm_table
*lkmtp, int cmd)
{
  if (cmd == LKM_E_LOAD) /* загрузка
модуля */
  {
    printf ("syshack loadedd\n");

    /* сохраняем адрес оригинального
вызова mkdir                         */
    (sy_call_t *)old_mkdir =
sysent[SYS_mkdir].sy_call;

   /* устанавливаем вместо mkdir свою
«заглушку»                           */
   sysent[SYS_mkdir].sy_call =
(sy_call_t *)hack;
  }

  if (cmd == LKM_E_UNLOAD) /* выгрузка
модуля                               */
  {
    printf ("syshack unloadedd\n");

    /* снимаем свою «заглушку»,
возвращая на место mkdir             */
    sysent[SYS_mkdir].sy_call=
(sy_call_t*)old_mkdir;
  }
  return(0);
}

/* точка входа в модуль               */
int entry(struct lkm_table *lkmtp,
int cmd, int ver)
```

```
{
  /* сердце модуля — макрос DISPATCH */
  DISPATCH(lkmtp, cmd, ver, load,
load, lkm_nofunc);
}
```

Сердцем модуля является макрос DISPATCH, передающий управление функции инициализации и деинициализации (в нашем случае называется load), вызываемой при загрузке и выгрузке модуля. Для перехвата/освобождения системного вызова mkdir используется прямая правка таблицы системных вызовов sysent. В принципе, можно было воспользоваться макросом MOD_SYSCALL, но это малоинтересно.

компиляция нашего LKM-модуля компилятором gcc

```
# gcc -D_LKM -D_KERNEL -I/sys -c syshack.c
```

Make-файл, собирающий LKM-модули

```
KSRCS=syshack.c
KOBJS=syshack.o

KMOD=syshack
CFLAGS= -D_LKM -D_KERNEL -I/sys
```

За загрузку модуля в память ядра отвечает утилита modload («man 8 modload»).

загрузка LKM-модуля в память ядра утилитой modload (hack — имя модуля в памяти, entry — точка входа в модуль, syshack.o — имя скомпилированного объектного файла)

```
# modload -o hack -eentry syshack.o
Module loaded as ID 0
```

Проверить успешность загрузки модуля можно утилитой «modstat» («man 8 modstat»).

утилита modstat показывает наличие модуля hack в памяти — значит, загрузка прошла успешно

```
# modstat
Type      Id Off Loadaddr Size Info
Rev Module Name
SYSCALL    0 210 e0b92000 0002 e0b93008
2 hack
```

Если модуль действительно загружен, то появится строчка с его именем (в данном случае — «hack»), и с этого момента любые попытки создать новый каталог утилитой mkdir будут обречены на провал, вплоть до того времени, пока не выгрузим модуль из памяти утилитой modunload («man 8 modunload»).

выгрузка модуля из памяти утилитой modunload

```
# modunload -n hack
```

Перехват остальных системных вызовов осуществляется аналогично. Таким образом, модуль может скрывать от глаз администратора некоторые процессы или файлы, «стелсируясь» на уровне ядра. А вот в FreeBSD модули реализованы совсем иначе...

➔ **KLD-модули FreeBSD.** Ранние версии FreeBSD поддерживали LKM-модули наравне со своими конкурентами, но, начиная с FreeBSD 3.0, интерфейс модулей был изменен на KLD, что расшифровывается как Dynamic Kernel Linker — динамическое связывание ядра. И LKM-модули отошли на задний план (в текущих версиях FreeBSD их поддержка прекращена).

В практическом плане это, в первую очередь, означает, что старые исходные тексты необходимо переделывать, а в некоторых случаях — чуть ли не переписывать полностью заново. На этом фоне преимущества нового типа модулей полностью девальвируются. Кстати говоря, штатное руководство («man KLD») лишь заявляет о преимуществах, но не перечисляет их, и за разъяснением приходится обращаться к другим источникам.

Если не углубляться в детали, то LKM-модуль — это ELF-файл, загружаемый в адресное пространство ядра, а KLD — это часть самого ядра, которая, в отличие от LKM, может быть загружена в любое время без поддержки прикладного уровня. То есть ядро в процессе старта системы как бы собирает себя из блоков, загружаемых/выгружаемых в любой момент времени.

KLD-модули предоставляют больше возможностей для разработчиков драйверов, но на нас это никак не распространяется. Перехват системных модулей реализуется так же, как и раньше. Меняется только декларация модуля, макросы и некоторые структуры данных.

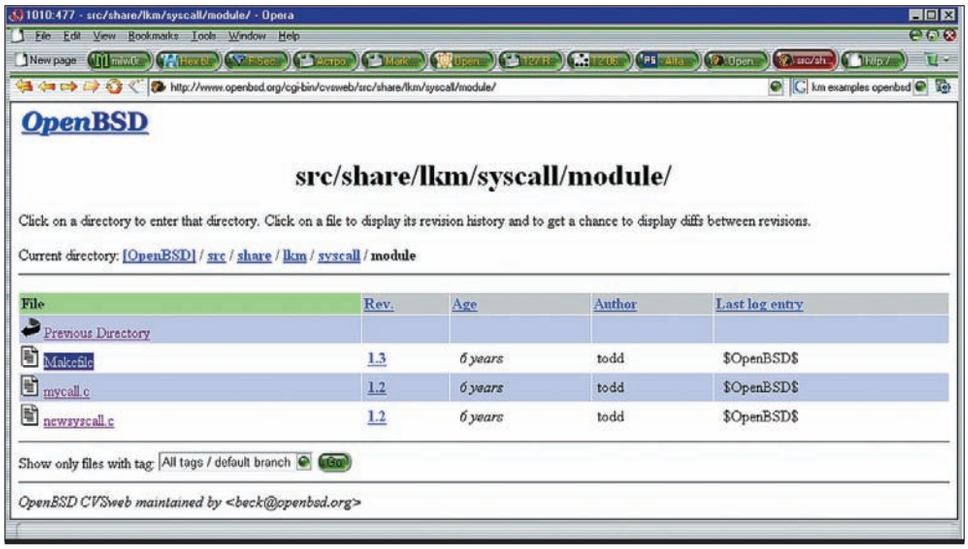
Примеры готовых модулей можно найти в каталоге /usr/share/examples/kld/ или, опять же, стянуть их из сети: **www.freebsd.org/cgi/cvsweb.cgi/src/share/examples/kld/**. «Зрелость» файлов варьируется от нескольких месяцев до 7 (!) лет.

техника перехвата системного вызова под FreeBSD из KLD-модуля

```
/* модуль, перехватывающий mkdir,
и работающий под FreeBSD                */
/* based on:                             */
/* syscall.c by Assar Westerlund
and hacked_mkdir.c by Joseph Kong       */
#include <sys/types.h>
#include <sys/param.h>
#include <sys/proc.h>
#include <sys/module.h>
#include <sys/sysent.h>
#include <sys/kernel.h>
#include <sys/sysproto.h>
#include <sys/systm.h>
#include <sys/syscall.h>

/* функция-«заглушка», устанавливаемая
на место mkdir                          */
static int hack (struct proc *p, void *arg)
{
  printf ("rock you!\x7\n"); return 0;
}

/* элемент структуры sysent,
описывающий наш системный вызов         */
static struct sysent hack_sysent = {
  1,        /* sy_narg */
  hack      /* sy_call */
};

/* процедура начальной загрузки модуля*/
static int load (struct module *module,
int cmd, void *arg)
{
  int error = 0;

  switch (cmd)
  {
  case MOD_LOAD: /* загрузка модуля  */
    printf ("syshack loadedd\n");
```

```
Stop in /usr/share/examples/kld/dyn_sysctl.
*** Error code 1

Stop in /usr/share/examples/kld.
# ls
Makefile        cdev            dyn_sysctl      syscall
# cd cdev
# ls
Makefile        README          module          test
# cd module
# ls
@               cdev.h          cdev.o          machine         setdef1.c
Makefile        cdev.kld        cdevmod.c       setdef0.c       setdef1.o
cdev.c          cdev.ko         cdevmod.o       setdef0.o       setdefs.h
# kldload ./cdev.kld
kldload: Unsupported file type
kldload: can't load ./cdev.kld: Exec format error
# kldload ./cdev.ko

Sample Loaded kld character device driver
Copyright (c) 1998
Rajesh Vaidheeswarran
All rights reserved
#
```

Откомпилированный и загруженный демонстрационный KLD-модуль, представляющий собой драйвер символьного устройства



Демонстрационные примеры LKM-модулей, входящие в состав OpenBSD, последний раз модифицировались 6 лет назад!

```
    /* устанавливаем вместо mkdir свою
«заглушку»                                */
    sysent[SYS_mkdir]=hack_sysent;
    break;

  case MOD_UNLOAD: /* выгрузка модуля */
    printf ("syshack unloadedd\n");

    /* снимаем свою «заглушку»,
возвращая на место mkdir                  */
    sysent[SYS_mkdir].sy_call=
(sy_call_t*)mkdir;
    break;

  default:
    error = EINVAL;
    break;
  }
  return error;
}

/* структура, описывающая основные
параметры модуля                          */
static moduledata_t syscall_mod = {
  "Intercept",
  load,
  NULL
};

/* сердце программы — макрос
DECLARE_MODULE, декларирующий модуль  */
DECLARE_MODULE(syscall, syscall_mod,
SI_SUB_DRIVERS, SI_ORDER_MIDDLE);
```

Базовая процедура load практически никак не изменилась (поменялись лишь определения cmd-кодов), а вот в декларации модуля произошли большие перемены. Функция с макросом DISPATCH исчезла, а вместе с ней исчезла и необходимость указывать точку входа в модуль при его загрузке в память ядра. Новый макрос DECLARE_MODULE не только задает точку входа в модуль вместе с его типом, но также определяет порядок загрузки! Вообще-то, этот макрос — не единственный, и с не меньшим успехом мы могли бы воспользоваться DEV_MODULE или SYSCALL_MODULE («man 9 DEV_MODULE» и «man 9 SYSCALL_MODULE»), но это уже дело вкуса, споры о котором рискуют превратиться в священные войны. А ведь прежде, чем воевать, модуль еще откомпилировать надо!

В общем случае сборка осуществляется следующим make-файлом, причем строки KO и KLMOD не являются обязательными:

```
SRCS   = syshack.c
KMOD   = syshack
KO     = ${KMOD}.ko
KLDMOD   = t

.include <bsd.kmod.mk>
```

Если компиляция прерывается сообщением «can't find kernel source tree», то это значит, что у тебя не установлены исходные тексты ядра, или bsd.kmod.mk-файл не может их найти. Установить недостающие компоненты можно в любой момент, запустив утилиту /stand/sysinstall и отметив пункт «Kernel Developer — Full binaries and doc, kernel source only». Выходит, чтобы откомпилировать KLD-модуль, необходимо иметь сырцы ядра! Вот такая она, FreeBSD! Ни NetBSD, ни OpenBSD ничего подобного не требуют (что вполне логично: LKM-модули, в отличие от KLD, не являются частью ядра).

После компиляции на диске образуется множество «левых» файлов и ссылок на системные каталоги (которые можно тут же удалить), среди которых затерялся файл с расширением .ko — это и есть наш модуль (в данном случае он называется syshack.ko).

Загрузка модуля в память осуществляется утилитой kldload («man 8 kldload»), которой указывается имя модуля (если модуль расположен в текущей директории, то необходимо предварить его ./), и, при желании, ключ -v — для более жестокой проверки корректности модуля.

Убедиться в успешности загрузки поможет утилита kldstat («man 8 kldstat»), которая, будучи запущенная без аргументов, выводит «свиток» всех имеющихся модулей. Если среди них присутствует syshack.ko, то операция перехвата прошла успешно, и теперь всякая попытка создания новой директории будет обречена на провал. Вплоть до выгрузки модуля из памяти, что можно сделать в любое время утилитой kldunload («man 8 kldunload»), указав ей имя модуля без расширения.

полный протокол перехвата и освобождения системного вызова mkdir посредством KLD-модулей

```
# kldstat        ; запускаем kldstat,
чтобы просмотреть список модулей
Id Refs Address     Size     Name
 1  3 0xc0100000 394090   kernel    ; ядро
 2  1 0xc0cac000 3000   daemon_saver.ko
; хранитель экрана
 3  1 0xc0caf000 14000  linux.ko
; эмулятор LINUX'a
; как видно, syshack-модуля среди них нет
(было бы удивительно, если бы он был)

# ls
; просматриваем текущий каталог утилитой ls
Makefile syshack.c syshack.ko syshack.o
; файл syshack.ko — это и есть
откомпилированный KLD-модуль

# kldload ./syshack
; загружаем наш модуль в память
syshack loaded
# Jun 22 13:58:20  /kernel: syshack loadedd
Jun 22 13:58:20  /kernel: syshack loadedd
; модуль рапортует об успешной загрузке,
; и система дублирует это сообщение,
указывая время его появления

# kldstat        ; снова просматриваем
список загруженных модулей
Id Refs Address     Size     Name
 1  4 0xc0100000 394090 kernel ; ядро
 2  1 0xc0cac000 3000 daemon_saver.ko
; хранитель экрана
 3  1 0xc0caf000 14000 linux.ko
; эмулятор LINUX'a
10  1 0xc08e3000 2000 syshack.ko
; вот он, наш модуль!
; как видно, syshack появился в списке
модулей,
; значит, загрузка и перехват
системного вызова mkdir прошли успешно
```

```
# mkdir TEST-DIR
; пытаемся создать каталог TEST-DIR
rock you! ; сообщение нашего модуля
# Jun 22 13:58:57  /kernel: rock you!
Jun 22 13:58:57  /kernel: rock you!
; ...но вместо создания нового каталога
; mkdir пищит спикером и посылает
нас на хутор за бабочками!

# ls ; просматриваем текущий каталог
Makefile syshack.c syshack.ko syshack.o
; директории TEST-DIR действительно нет,
; вот что значит правильно
организованный перехват!

# kldunload syshack
; выгружаем модуль из памяти
syshack unloadedd
# Jun 22 14:00:44 /kernel: syshack unloadedd
Jun 22 14:00:44 /kernel: syshack unloadedd
; модуль выгрузил себя из памяти,
; восстановив оригинальный mkdir

# mkdir TEST-DIR
; пытаемся создать TEST-DIR еще раз
; теперь на экран не выводится
никаких сообщений

# ls
; проверяем успешность создания TEST-DIR
Makefile TEST-DIR syshack.c  syshack.ko
syshack.o
; каталог TEST-DIR действительно создан!
; значит, mkdir был восстановлен правильно!

# kldstat ; просматриваем список модулей
Id Refs Address     Size     Name
 1  3 0xc0100000 394090 kernel ; ядро
 2  1 0xc0cac000 3000 daemon_saver.ko
; хранитель экрана
 3  1 0xc0caf000 14000 linux.ko
; эмулятор LINUX'a
; модуля syshack в этом списке нет,
; значит, его выгрузка прошла успешно
```

➔ работа с libkmv, когда модули недоступны. Все операционные системы семейства BSD поддерживают библиотеку libkvm (Kernel Virtual Memory), предоставляющую унифицированный доступ к памяти ядра, как KLM-модули, но в отличие от них, сохранившую полную обратную совместимость. Другими словами, программа, написанная для FreeBSD, при переносе на OpenBSD или NetBSD не потребует никаких изменений!

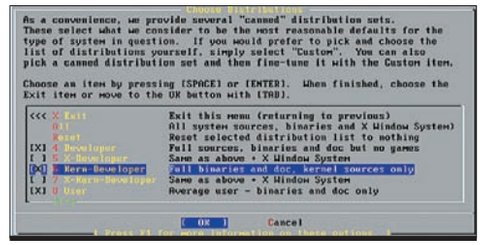
Библиотека libkvm представляет собой высокоуровневую обертку вокруг псевдоустройства /dev/mem, изображающего из себя физическую оперативную память (псевдоустройство /dev/kmem включает в себя лишь виртуальную память ядра после трансляции адресов). Аналогичное псевдоустройство имеется и в LINUX'e, но соответствующей библиотеки для него нет, что жутко напрягает. Тем не менее, с псевдоустройством /dev/[k]mem на всех системах можно работать и напрямую, через обычный ввод/вывод, для обеспечения полной переносимости. Однако целесообразность этого решения весьма сомнительна, поэтому сосредоточимся исключительно на библиотеке libkvm, а остальные способы доступа к ядерной памяти оставим за кадром.

Прежде чем работать с виртуальной памятью ядра, ее необходимо открыть, вызвав функцию kvm_open («man kvm_open») и передав ей в качестве имени файла NULL. Тогда, при успешном завершении операции, обеспеченная правами root'a, она вернет дескриптор. Если вместо NULL указать имя файла-образа ядра или кору, то открыты будут они, а не «живое» ядро в памяти, но нам это не нужно.

Передавая полученный дескриптор функциям kvm_read и kvm_write, мы сможем читать/писать память по заданным виртуальным адресам. Но какие именно адреса мы хотим читать? Вернее, как найти среди множества адресов полезную информацию, например таблицу системных вызовов? В этом поможет функция kvm_nlist, разбирающая таблицу символов и возвращающая адрес элемента по его имени. Единственным ее аргументом (не считая дескриптора памяти ядра) является указатель на массив структур nlist, описанных в одноименном включаемом файле. В поле n_name заносится имя интересующего элемента, и если этот элемент действительно присутствует в таблице символов, то после завершения функции в поле n_value возвращается его виртуальный адрес.

Основные способы проникновения в ядро в различных BSD-системах

	FreeBSD	NetBSD	OpenBSD
ядро	монолитное с модулями	монолитное с модулями	монолитное (иногда с модулями)
тип модулей	KLD	LKM	LKM
загрузка модуля	kldload	modload	modload
выгрузка модуля	kldunload	modunload	modunload
статистика по модулям	kldstat	modstat	modstat
исходные тексты ядра	не требует	не требует	не требует
интерфейс kvm	поддерживается	поддерживается	поддерживается



Компиляция KLD-модулей требует наличия исходных текстов ядра, которые легко установить, запустив утилиту /stand/sysinstall

Приведенная ниже программа (любезно позаимствованная из статьи «Playing Games With Kernel Memory... FreeBSD Style», опубликованной в #63 PHACK'e) определяет адрес таблицы системных вызов, адрес «нашего» системного вызова и адрес функции, по которой данный вызов располагается в памяти. Программа требует два аргумента: имя системного вызова (например, mkdir) и его номер (в случае mkdir равный 1), рекомендуя обратиться к файлу /usr/src-/sys/sys/syscall.h, если номер вызова не известен (вообще-то данный файл располагается в каталоге /usr/include/sys/, но это неважно). На самом деле, имя системного вызова используется только для того, чтобы контролировать его существование. Для вычисления адреса используется номер syscall'a, который преобразуется в индекс таблицы системных вызовов. Это грубая недоработка! Если мы успешно определи адрес syscall'a по имени, то зачем нам его номер?! Если же мы можем (а мы можем) определять адреса syscall'ов по номеру через индекс в таблице системных вызовов, то зачем нам нужно имя?!

определение виртуальных адресов системных вызовов на FreeBSD, NetBSD и OpenBSD

```
/* программа, демонстрирующая технику
определения адресов системных вызовов, */
/* работающая на всем зоопарке
BSD-подобных систем                    */
/* Based on Stephanie Wehner's
checkcall.c,v 1.1.1.1                  */
#include <stdio.h>
#include <fcntl.h>
#include <kvm.h>
#include <nlist.h>
#include <limits.h>
#include <sys/types.h>
#include <sys/sysent.h>
#include <sys/syscall.h>

int main(int argc, char *argv[])
{
  char errbuf[_POSIX2_LINE_MAX];
  kvm_t *kd; u_int32_t addr; int
callnum; struct sysent call;
  struct nlist nl[] = { { NULL },
{ NULL }, { NULL }, };

  if(argc != 3)
  {
```

```
    printf("Usage:\n%s <name of syscall>
<syscall number>\n\n", argv[0]);
    printf("See /usr/src/sys/sys/syscall.h
for syscall numbers\n");exit(0);
  }

  /* Find the syscall */
  nl[0].n_name = "sysent"; nl[1].n_name
= argv[1]; callnum = atoi(argv[2]);

  /* Initialize kernel virtual memory
access */
  kd = kvm_openfiles(NULL, NULL, NULL,
O_RDWR, errbuf);

  /* Find the addresses */
  kvm_nlist(kd, nl);

  if(!nl[0].n_value)
    return fprintf(stderr,"ERROR: %s
not found\n", nl[0].n_name);
  else
    printf("%s is 0x%x at 0x%x\n",nl[0].
n_name,nl[0].n_type,nl[0].n_value);

  /* Calculate the address */
  addr = nl[0].n_value + callnum *
sizeof(struct sysent);

  /* Print out location */
  if(kvm_read(kd, addr, &call,
sizeof(struct sysent)) < 0)
    return fprintf(stderr, "ERROR:
%s\n", kvm_geterr(kd));
  else
    printf("sysent[%d] is at 0x%x
and will execute function"
      " located at 0x%x\n", callnum,
addr, call.sy_call);

  kvm_close(kd);
}
```

При трансляции листинга компилятору необходимо указать на библиотеку libkvm (при этом «lib», как всегда, опускается), иначе линкер начнет материться на неразрешимые ссылки.

компиляция программы find_syscall.c

```
#gcc find_syscall.c -o find_syscall -lkvm
```

Откомпилировав программу, попробуем определить адрес системного вызова mkdir. На тестируемой машине (FreeBSD 4.5) результат выглядит так:

```
#./find_syscall mkdir 1
Finding syscall 1: mkdir

sysent is 0x4 at 0xc03ca480
sysent[1] is at 0xc03ca488 and will
execute function located at 0xc01ba2cc
```

Воспользовавшись функцией kvm_write, без труда поменяем указатель на mkdir в таблице системных вызовов или внедрим jump в начало самой mkdir (но последний способ не очень надежен и даже на однопроцессорных машинах может приводить к сбоям, поскольку существует вероятность, что правка функции совпадет с ее вызовом).

Остается решить последний вопрос: куда перенаправлять перехваченный системный вызов. На пользовательское адресное пространство — нельзя, система таких шуток не понимает. Теоретически, можно найти свободное место в ядре (заполненное, например, NOP'ми), записав в него крошечный «бустер», выделяющий немного памяти через malloc для размещения основного кода перехватчика, который затягивается внутрь ядра через copyin. Но никакой гарантии, что свободное место найдется, нет, поэтому лучше (и надежнее!) размещать перехватчик поверх какого-нибудь редко используемого системного вызова, например, устаревшего, но до сих пор поддерживаемого lstat, проходящего под номером 40. Или SYS_ptrace/SYS_ktrace, «ослепив» кучу утилит, предназначенных для выявления вредоносных программ, что, в конечном счете, не помешает собственной маскировке.

➔ **перехват системных вызовов —** прерогатива не только зловредных программ. Тем же самым занимаются и средства защиты, активно использующие интерфейс kvm, который поддерживает даже суперзащищенная OpenBSD. И вообще, следует различать действие и его мораль. А мораль такова, что распространенность BSD-систем создает все предпосылки для локальных и удаленных

UNIX-подобным системам приходится конкурировать не только с Windows, но и воевать между собой

атак с применением всех доступных средств и интерфейсов. Главное — знать как. Все остальное — дело техники и... фантазии. В модификации ядра есть свое непередаваемое очарование, притягивающее, словно магнитом, и заставляющее рыскать в поисках скудной документации по всей сети, перечитывать man и, конечно же, экспериментировать!

Проблема в том, что код, работающий на одной системе, может оказаться совершенно неработоспособным на другой. Поэтому желательно иметь в своем распоряжении хотя бы по одной версии каждой из BSD-систем. Для этой цели хорошо подходят виртуальные машины типа VM Ware. Дисковое пространство давно перестало быть проблемой, а в нормальной конфигурации (то есть без иксов) BSD-системы свободно умещаются в половину гигабайта — смехотворная по нынешним временам величина C

```
# ls
Makefile        syshack.c       syshack.ko      syshack.o
# kldload ./syshack
syshack loadedd
# Jun 22 13:47:54  /kernel: syshack loadedd
Jun 22 13:47:54  /kernel: syshack loadedd

# kldstat
Id Refs Address    Size     Name
 1    4 0xc0100000 394090   kernel
 2    1 0xc0cac000 3000     daemon_saver.ko
 3    1 0xc0caf000 14000    linux.ko
 8    1 0xc08e3000 2000     syshack.ko
# mkdir XXX
rock you!
# Jun 22 13:48:08  /kernel: rock you!
Jun 22 13:48:08  /kernel: rock you!

# ls
Makefile        syshack.c       syshack.ko      syshack.o
# kldunload syshack
syshack unloadedd
# Jun 22 13:49:07  /kernel: syshack unloadedd
Jun 22 13:49:07  /kernel: syshack unloadedd

# mkdir XXX
# ls
Makefile        XXX             syshack.c       syshack.ko      syshack.o
#
#
```

Загрузка и выгрузка KLD-модуля

ЭНЦИКЛОПЕДИЯ GamePost

Незаменимый помощник при выборе игры

Описание:

Основанная на классическом фильме 1972 года, The Godfather, эта игра погрузит вас в опасный мир мафии. Создайте своего персонажа и проживите вместе с ним 10 лет – с 1945 по 1955 год – в семье Корлеоне. Вас ждут геймплей в стиле GTA, новая сюжетная линия и озвучка от актеров фильма, включая Марлона Брандо.

The Godfather	Жанр:
$69.99	**Action**

Описание:

Factions – это дополнение к знаменитой MMORPG Guild Wars, славящейся полным отсутствием абонентской платы. Оно включает в себя новую кампанию, новые регионы, профессии, способности, задания и врагов, а также расширенные возможности для гильдий и PvP.

Guild Wars: Factions (EURO)	Жанр:
$69.99	**RPIG**

Описание:

Безумно красивая, традиционно нелинейная и невероятно реалистичная – The Elder Scrolls IV: Oblivion являет собой эталон для последующих ролевых игр. Более 1000 NPC, каждый с полноценной озвучкой и анимацией, населяют огромный мир Oblivion – они живут своими жизнями 24 часа 7 дней в неделю, едят, спят, работают.

Elder Scrolls IV Oblivion Collector's Edition	Жанр:
$79.99	**Role Playing**

САМАЯ ПОЛНАЯ ИНФОРМАЦИЯ ОБ ИГРАХ

* Огромное количество скриншотов
* Исчерпывающие описания
* Возможность посмотреть внутренности коробок

Играй просто!
GamePost

Тел.: (495) 780-8825
Факс.: (495) 780-8824
www.gamepost.ru

GAMEPOST

пламенный почтовик

ВОЗВОДИМ БЕЗОПАСНЫЙ ПОЧТОВЫЙ СЕРВЕР

В СОВРЕМЕННЫХ КОРПОРАТИВНЫХ ОРГАНИЗАЦИЯХ ОСНОВНЫМ СПОСОБОМ ПЕРЕДАЧИ ИНФОРМАЦИИ МЕЖДУ СОТРУДНИКАМИ ВНУТРИ ОФИСА ПО-ПРЕЖНЕМУ ОСТАЕТСЯ ОБМЕН ЭЛЕКТРОННЫМИ ПОЧТОВЫМИ СООБЩЕНИЯМИ

ВОЛЬФ Д. А. AKA PAYHASH

На сегодняшний день этот метод обмена электронными данными представляет собой наибольшую угрозу из-за возможности раскрытия информации третьим лицам и, соответственно, может сыграть необратимую роль в судьбе организации. Механизму предотвращения подобных ситуаций и посвящена эта небольшая статья.

Каждый уважающий себя администратор (естественно, в меру своих знаний) не обходит стороной такие вещи, как SSL(Secure Sockets Layer), IPSec (Internet Protocol Security) и т.д., поскольку эти понятия являются базовыми для защиты информации. К сожалению, не все организации способны содержать в штате и администратора, и специалиста по сетевой защите информации, поэтому многим админам приходится совмещать эти должности. Следовательно, на него ложится двойная ответственность. И чтобы помочь нашему читателю избежать связанного с этой ответственностью небольшого нагоняя от начальства, разберем один из методов на примере настройки корпоративного почтового сервера с поддержкой шифрования данных.

→ **в бой!** Начнем мы с установки и настройки криптомеханизма OpenSSL (**www.openssl.org**), поскольку этот механизм свободен в использовании, не требует каких-либо дополнительных соглашений (кроме GPL) и ни в чем не уступает своим платным коллегам. Скачиваем исходные коды с последним стабильным дистрибутивом OpenSSL с ресурса **www.openssl.org/source** (кстати, не забудь вставить в дисковод наш диск — там присутствует весь необходимый для статьи софт). Также можно позволить себе скачать последнюю LATEST-версию **www.openssl.org/source/openssl-0.9.8b.tar.gz**. Скачал? Теперь распаковывай дистрибутив в любой подходящий каталог. Например, в /tmp/sandbox:

```
terminal# mkdir /tmp/sandbox
terminal# cd /tmp/sandbox/
terminal# wget http://www.openssl.org/
```

```
source/openssl-0.9.8b.tar.gz
Распознается www.openssl.org... 195.30.6.166
Connecting to www.openssl.org
|195.30.6.166|:80... соединение установлено
Запрос HTTP послан, ожидается ответ... 200 OK
Длина: 3,279,283 (3.1M) [application/x-tar]
Загружено... 100%
17:43:43 (33.16 KB/s) — `openssl-
0.9.8b.tar.gz' saved [3279283/3279283]
terminal# tar -zxvf /tmp/sandbox/-
openssl-0.9.8b.tar.gz
```

Привычным образом конфигурируем, компилируем и устанавливаем программу:

```
terminal# cd /tmp/sandbox/openssl-0.9.8b
terminal# ./config
terminal# make all
terminal# make test
terminal# make install
terminal# rehash
```

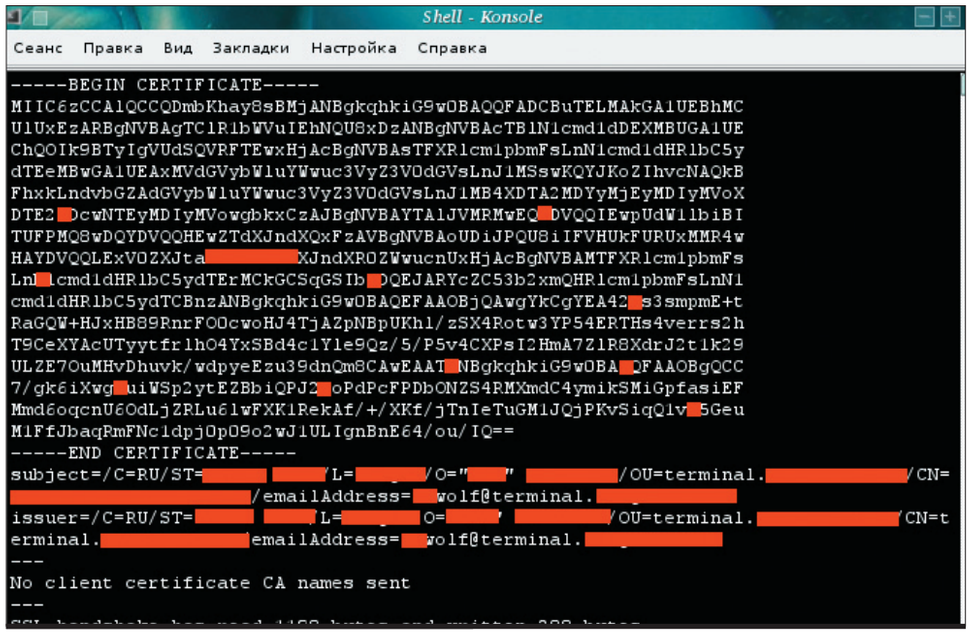
С установкой OpenSSL разобрались, что не может не радовать. А что же все-таки такое OpenSSL? Проще говоря, это комплекс (набор программ и библиотек) по управлению механизмом криптографии. Сюда входят функции и программы, которые обеспечивают основные алгоритмы и методы для шифрования. Большинство GPL-программ, таких как MySQL, Sendmail, OpenLDAP и т.д. используют библиотеки, которые предоставляет пакет OpenSSL. С помощью программ, которые предоставляет комплекс OpenSSL, мы сможем управлять механизмом закрытых и открытых ключей, а также обеспечивать ЭЦП и выпускать сертификаты. Уверен, что с теорией симметричного и ассиметричного шифрования знакомы все, поэтому не будем заниматься скучными рассказами про Алису и Боба, которые пишут друг другу секретные письма эротического содержания, а погрузимся в практику. Если мы хотим создать пару с закрытым и открытым ключом, то в OpenSSL это делается так:

```
openssl genrsa  -des3 -rand /usr/local/-
games/doom3-linux-1.3.1302.x86.run -out
sendmail.key 1024
```

Эта команда сгенерирует секретный 1024-битный ключ RSA, где опция -rand — это входные данные, необходимые для того, чтобы получить надежный псевдослучайный набор простых чисел. Также нам необходимо будет ввести ключевое слово — пароль на секретный ключ, который нужно запомнить. Итак, ключ, с помощью которого мы сможем дешифровать данные, создан. Назначим на него права, которые будут беречь его от посторонних глаз:

```
terminal# chmod 700 sendmail.key
```

Конечно же, мы не зря выбрали такое имя ключа — это поможет нам избежать путаницы.



openssl s_client показывает, что smtp протокол защищен сертификатом

На основе нашего секретного ключа создадим его публичную часть. Сделать это можно следующим образом:

```
terminal# openssl rsa -in sendmail.key
-out pubsendmail.pem -pubout
```

Для того чтобы зашифровать какое-либо текстовое сообщение при помощи публичного ключа, необходимо выполнить следующую команду:

```
terminal# openssl rsautl -in file.txt -
out file.crypt -inkey pubsendmail.pem -
pubin -encrypt
```

Здесь rsautl — это алгоритм шифрования, in file.txt — файл, нуждающийся в шифровании (его подаем на вход), out file.crypt — шифрованный файл, который будет получаться на выходе, inkey pubsendmail.pem — ключ, с помощью которого мы будем шифровать файл file.txt, a pubin и encrypt — аргументы, которые сообщают программе rsault, что надо совершить шифрование с помощью открытого ключа.

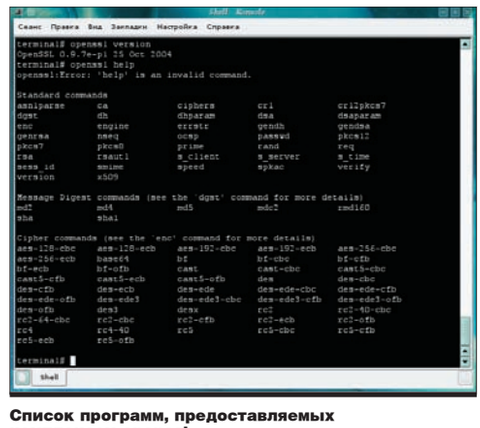
В результате этой команды из файла file.txt будут извлечены чистые данные, которые программа rsault зашифрует и выдаст результат в файл file.crypt. Для того чтобы дешифровать файл file.crypt с помощью OpenSSL нужно совершить следующее:

```
terminal# openssl rsautl -in file.crypt
-out file.decrypt -inkey server.key -
decrypt
```

Файл file.decrypt содержит дешифрованный текст, который был в исходном состоянии в файле file.txt.

Итак, мы разобрались с базовым понятием криптографии при помощи шифрования с использованием открытого ключа.

➔ **сертификаты.** Представь, что наш открытый ключ подписала некая доверительная (сторонняя) организация. Теперь такой открытый ключ будет являться не просто ключом, а целым сертификатом, удостоверяющим пользователей в том, что данный ключ несет полную гарантию целостности зашифрованных данных и является настоящей собственностью владельца сертификата. Это и есть понятие сертификата. Иначе говоря, существует некий публичный ключ и его подписал некий доверительный центр (своим дайджестом или чем-либо еще). Данный ключ действительно является публичным ключом владельца, и после подписания он превращается в доверительный сертификат и может быть использован. Для того чтобы правильно сформировать открытый ключ, из которого мы будем формировать сертификат (в OpenSSL этим занимается программа req), нужно сделать следующее:



Список программ, предоставляемых комплексом openssl

```
terminal# openssl req -new -key sendmail.key -out sendmail.csr
```

После ввода нашего секретного пароля на экран выпадет некий диалог, анкета, которую необходимо будет заполнить, после чего в файле sendmail.csr будет записана информация, взятая из заполняемой анкеты и часть приватного ключа в виде открытых данных (публичный ключ).

Затем файл sendmail.csr необходимо направить доверительному центру CA, где его удостоверят подписью CA на основе дайджеста корневого сертификата и произведут кое-какие изменения, после чего и получится наш сертификат. Стоит отметить, что данная процедура не бесплатна: стоит она примерно 300 американских президентов (такова платная сторона бесплатного GPL, подробнее читаем Крис Касперски «Платная сторона бесплатного GPL»). Но не стоит отчаиваться: не все в этом мире строится на деньгах и уважении. Почему бы самому не подписать свой сертификат? Правда, это будет не совсем то, но работать будет. Таким образом, мы подошли к понятию самоподписанных сертификатов. Такие сертификаты подобны корневым сертификатам CA-центров.

Чтобы создать такой сертификат нужно файл sendmail.crt подписать дайджестом на основе секретного ключа. Это может сделать программа x509:

```
terminal# openssl x509 -req -days 3666 -in sendmail.csr -signkey sendmail.key -out sendmail.crt
```

Вот так мы создадим сертификат в виде файла sendmail.crt, который будет действителен 3666 дней. Теперь для нас важны два файла: sendmail.key и sendmail.crt. На них мы и будем базировать криптоподдержку в службах.

Группы, работающие в сетевой безопасности, различают несколько уровней шифрования данных, передаваемых по сети. Они начинаются на сетевом уровне IP и заканчиваются седьмым уровнем приложений. Мы же рассмотрим шифрование на транспортном уровне, называемое SSL/TLS (TLS — это SSL v3.1), так как данный уровень подходит почти ко всем приложениям и является для них прозрачным. Программа sendmail умеет работать на этом уровне, то есть она способна предоставлять клиентам шифрованную передачу электронных почтовых сообщений посредством SMTP-трафика.

Подразумевается, что некоторый опыт установки sendmail у тебя уже есть (в крайнем случае, документацию всегда можно найти в internet), поэтому некоторые моменты, относящиеся к настройке sendmail мы не будем комментировать.

Итак, закачиваем свежую стабильную версию sendmail в уже известный каталог /tmp/sandbox и распаковываем trball.

```
terminal# cd /tmp/sandbox/
terminal# wget \ ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.13.7.tar.gz
tar -zxvf sendmail.8.13.7.tar.gz
terminal# tar -zxvf sendmail.8.13.7.tar.gz
```

Заходим в каталог sendmail-8.13.7/devtools/Site/:

```
terminal# cd sendmail-8.13.7/devtools/Site/
```

В каталоге создаем файл site.config.m4 на редактирование:

```
terminal# ee site.config.m4
```

В файл site.config.m4 вписываем следующие строчки:

```
dnl Stuff for TLS
APPENDDEF(`confINCDIRS', `-I/usr/local/include')
APPENDDEF(`confLIBDIRS', `-L/usr/local/lib')
APPENDDEF(`conf_sendmail_ENVDEF', `-DSTARTTLS')
APPENDDEF(`conf_sendmail_LIBS', `-lssl -lcrypto')
```

Сохраняем файл site.config.m4 и выходим из него. Если планируется прикрутить к sendmail антиспам-

SPECIAL МНЕНИЕ

АНТОН КАРПОВ

Специалист в области информационной безопасности. Круг профессиональных интересов: сетевые атаки, безопасность UNIX-систем, безопасность беспроводных сетей...

ДЕЙСТВИТЕЛЬНО ЛИ ПРАВОМЕРНО ЗАКЛЮЧЕНИЕ, ЧТО BSD И ДЕСКТОП СОВЕРШЕННО НЕСОВМЕСТИМЫ?

Проблема в том, что понимать под «десктопом», ведь четкого определения и описания функционала у этого термина нет. Для кого-то «десктоп» — это пишущая машинка, чтобы серфить интернет, читать почту и составлять документы. Для кого-то — мультимедийная станция для игрушек или графической/музыкальной работы. Кто-то использует систему на обычном домашнем писюке, из периферии пользуясь только клавиатурой и мышкой, а кто-то имеет навороченный ультрасовременный ноутбук и жить не может без bluetooth, irda и wifi- коннективити. Для всех этих людей понятия и требования к термину десктоп различны.

Поэтому, не отвечая напрямую на вопрос, проведу сравнение с дистрибутивами Linux, ведь апологеты этой ОС давно утверждают, что Linux «готов для десктопа не хуже винды». Во-первых, и на Linux, и на BSD есть возможность запускать одинаковый набор десктоп-приложений. KDE, Gnome, Gimp, OpenOffice, xmms, mplayer, psi, и многое другое — все это работает как на Linux, так и на BSD. То есть, с точки зрения набора софта, отличий нет. Что касается поддержки таких «десктопных» технологий как ACPI, Bluetooth, то здесь конкуренцию Linux может составить разве что FreeBSD, так как в остальных *BSD эти подсистемы находятся в зачаточном состоянии. Хуже ситуация обстоит и с полноценной поддержкой графических адаптеров: драйвера для устройств от Nvidia выпускаются только для Linux и FreeBSD, для карточек от ATI — только под Linux. Впрочем, многим аппаратное ускорение и другие фишки не особо нужны для повседневной работы.

фильтр, антивирусный пакет или и то и другое, тогда в тот же файл site.config.m4 необходимо добавить строчку с поддержкой мильтера:

```
#MILTER
APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')
```

А теперь переходим в каталог ../../libmilter/ и собираем библиотеки с поддержкой мильтера:

```
terminal# cd ../../libmilter/
terminal# ./Build -c
```

Далее заходим в каталог ../cf/cf/ и создаем в нем файл:

```
sendmail.mc:
terminal# cd ../cf/cf/
terminal# ee sendmail.mc
```

В файл sendmail.mc (подразумевается, что у тебя уже есть опыт установки sendmail из исходных кодов) добавляем строчки:

```
define(`confCACERT_PATH',
`/etc/mail/certs')
define(`confCACERT',
`/etc/mail/certs/sendmail.pem')
define(`confSERVER_CERT',
`/etc/mail/certs/sendmail.crt')
define(`confSERVER_KEY',
`/etc/mail/certs/sendmail.pem')
define(`confCLIENT_CERT',
`/etc/mail/certs/sendmail.crt')dnl
define(`confCLIENT_KEY',
`/etc/mail/certs/sendmail.pem')dnl
```

Это и есть наш минимальный рабочий конфигурационный файл sendmail на языке m4:

```
divert(0)
VERSIONID(`$FreeBSD: src/etc/sendma-
il/freebsd.mc,v GO TO HELL')
OSTYPE(freebsd6)
DOMAIN(generic)
FEATURE(use_ct_file)
FEATURE(access_db,
`hash -o -T<TMPF> /etc/mail/access')
FEATURE(blacklist_recipients)
FEATURE(local_lmtp)
FEATURE(mailertable,
`hash -o /etc/mail/mailertable')
FEATURE(virtusertable,
`hash -o /etc/mail/virtusertable')
FEATURE(relay_based_on_MX)
define(`confCACERT_PATH',
`/etc/mail/certs')
define(`confCACERT',
`/etc/mail/certs/sendmail.pem')
define(`confSERVER_CERT',
`/etc/mail/certs/sendmail.crt')
define(`confSERVER_KEY',
`/etc/mail/certs/sendmail.pem')
define(`confCLIENT_CERT',
`/etc/mail/certs/sendmail.crt')dnl
define(`confCLIENT_KEY',
`/etc/mail/certs/sendmail.pem')dnl
dnl TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5
CRAM-MD5 PLAIN LOGIN')dnl
dnl define(`confAUTH_MECHANISMS',
`GSSAPI DIGEST-MD5 CRAM-MD5 PLAIN
LOGIN')dnl
dnl define(`confDEF_AUTH_INFO',
`/etc/mail/auth/auth-info')dnl
DAEMON_OPTIONS(`Name=MTA')dnl
DAEMON_OPTIONS(`Port=465,
Name=MTA-SSL, M=s')dnl
define(`confMAX_RCPTS_PER_MESSAGE', `10')
define(`confMAX_MESSAGE_SIZE', `1048576')
define(`confBIND_OPTS',
`WorkAroundBrokenAAAA')
define(`confMAX_MIME_HEADER_LENGTH',
`256/128')
define(`confNO_RCPT_ACTION',
`add-to-undisclosed')
define(`confPRIVACY_FLAGS',
`authwarnings,noexpn,novrfy')
MAILER(local)
MAILER(smtp)
```

Итак, создали конфигурационный файл и сохранили его. Далее конфигурационный файл sendmail (на языке m4) необходимо преобразовать в конфигурационный файл, который должна понимать программа sendmail. Делаем это так:

```
terminal# m4 ../m4/cf.m4 sendmail.mc >
sendmail.cf
```

Далее нужно инсталлировать созданный файл sendmail.cf. Исторически место назначения — каталог /etc/mail (и далее под понятием каталога sendmail будем подразумевать каталог /etc/mail). Вперед:

```
terminal# make install-cf CF=sendmail
```

Ну что же, приступим непосредственно к самой сборке и установке sendmail. Переходим из каталога cf/cf/ на два уровня выше и выполняем скрипт Build:

```
terminal# cd ../../
terminal# ./Build -c
Далее выполняем Build install:
terminal# ./Build install
```

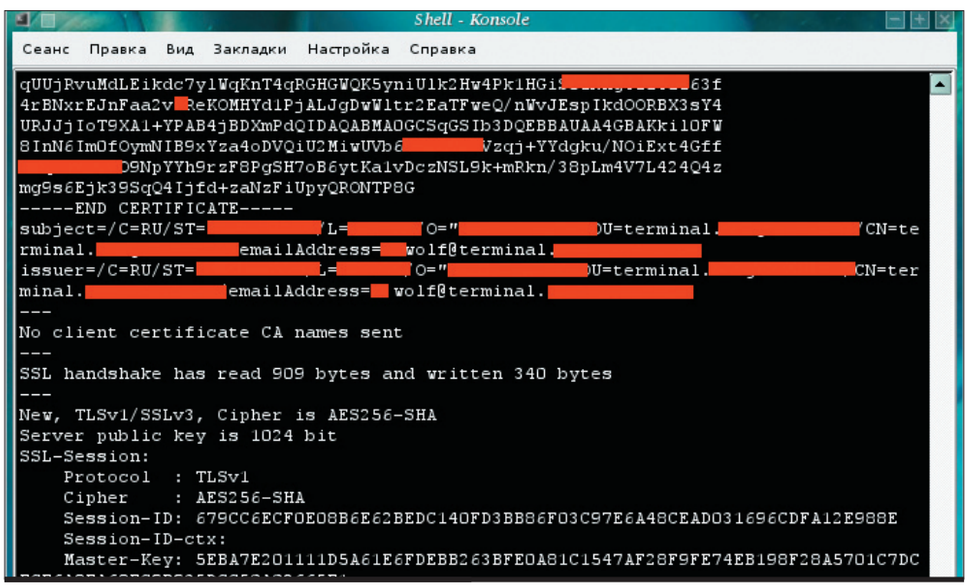
С установкой sendmail покончено. Теперь необходимо выполнить условия, описанные в конфигурационном файле относительно директив confCACERT_PATH,confCACERT,confSERVER_CERT,confSERVER_KEY, confCLIENT_CERT,confCLIENT_KEY. Создаем каталог /etc/mail/certs, копируем в него файлы sendmail.key и sendmail.crt (которые у нас уже имеются) и заходим в этот каталог:

```
terminal# mkdir /etc/mail/certs
terminal# cp /tmp/sandbox/sendma-
il.key/tmp/sandbox/sendmail.crt \
/etc/mail/certs/
terminal# cd /etc/mail/certs
```

Теперь нужно избавиться от пароля в секретном файле sendmail.key. Свершим это при помощи программы rsa:

```
terminal# openssl rsa -in sendmail.key
-out sendmail.pem
```

В результате мы получили секретный файл sendmail.pem. И я думаю, что закономерный вопрос,



pop3-протокол защищен сертефикатом

возникший у читателя относительно файла sendmail.pem в конфигурационном файле sendmail.mc, исчерпан.

Назначаем права 700 на каталог с сертификатами /etc/mail/certs, а файлы /etc/sandbox/sendmail.key, /etc/sandbox/sendmail.csr и /etc/sandbox/sendmail.crt нужно удалить:

```
terminal# chmod -R 700 /etc/mail/certs
terminal# cd /tmp/sandbox
terminal# rm sendmail.key sendmail.csr
sendmail.crt
```

Перед тем, как запускать sendmail, нам необходимо создать некоторых хэшей и просто файлов, необходимых для работы sendmail. Поэтому изволь почитать соответствующее руководство на нашем диске.

```
terminal# cd /etc/mail
```

Если файла aliases нет — создаем его:

```
terminal# touch aliases
```

Обязательно делаем newaliases:

```
terminal# newaliases
```

Создаем файл с доверительными и недоверительными хостами:

```
terminal# touch access
terminal# makemap hash access.db < access
terminal# touch virtusertable
terminal# makemap hash virtusertable.db
< virtusertable
terminal# makemap hash mailertable.db
< mailertable
terminal# touch local-host-names
```

Осталось только запустить sendmail и проверить наличие поддержки шифрования (если запустить sendmail не получается, то возможно, где-то допущена ошибка). Смотрим логи:

```
terminal# sendmail -bd -q10m
terminal# openssl s_client -host
localhost -port 465
```

➜ **5 минут — полет нормальный?** Если все прошло нормально, то программа s_client сможет подключиться на порт 465 (защищенный порт SMTP/SSMTP), а нам на экран выпадет сертификат сервера с публичным ключом. SMTP-службу мы защитили, осталось защитить службу POP, иначе теряется весь смысл защиты почтового сервера. В качестве POP-демона мы рекомендуем использовать программу popa3d, написанную русским хакером Solar Designer (Sergey Samoyloff). Программа является официальным дистрибутивом для FreeBSD и находится в портах FreeBSD /usr/ports/mail/popa3d.

В исходных кодах по умолчанию popa3d настроена для сборки под Linux-машину, поэтому нужно привести необходимые рекомендации по корректировке этого демона для BSD-систем. Итак, скачиваем последний дистрибутив popa3d с сайта Solar Designer'a **www.openwall.com/popa3d**. Распаковываем дистрибутив с «попой», заходим в каталог с его исходными кодами, открываем файл params.h на редактирование и находим строчки:

```
#define AUTH_PASSWD             0
#define AUTH_SHADOW             1
#define AUTH_PAM                0
#define AUTH_PAM_USERPASS       0
#define USE_LIBPAM_USERPASS     0
```

После чего приводим строчки в такой вид (сохраняем и выходим из редактора):

```
#define AUTH_PASSWD             1
#define AUTH_SHADOW             0
#define AUTH_PAM                0
#define AUTH_PAM_USERPASS       0
#define USE_LIBPAM_USERPASS     0
```

Теперь открываем файл Makefile на редактирование. Раскомментируем строчки: CFLAGS += -DHAVE_OPENSSL, LIBS += -lcrypt, LIBS += -lcrypto. Все, теперь можно компилировать и инсталлировать программу.

После того как мы поставили и запустили popa3d, нужно установить еще один программный продукт, который будет заниматься шифрованием POP3-трафика. Наиболее доступным и достаточно простым в использовании для организации SSL является программа Stunnel. Можно поставить ее непосредственно из исходных кодов, но лучше сделать это из программных портов FreeBSD: /usr/ports/security/stunnel.

Вот мы и установили stunnel без всякого физического напряжения.

Создаем конфигурационный файл /usr/local/etc/stunnel/stunnel.conf и открываем его на редактирование. В файл вписываем следующие строчки:

```
cert = /usr/local/etc/stunnel/stunnel.crt
key = /usr/local/etc/stunnel/stunnel.pem
;RNDfile = /usr/local/etc/stunnel/stunnel.rnd
chroot = /usr/local/var/stunnel/
setuid = stunnel
setgid = stunnel
pid = /run/stunnel.pid
output = /var/log/stunnel.log
ciphers = HIGH
debug = 6
compression = rle
[pop3s]
accept  = 995
connect = 127.0.0.1:110
```

А теперь создаем chroot-директорию и устанавливаем соответствующие права:

```
terminal# cd /usr/local/var && mkdir
stunnel && cd stunnel
terminal# mkdir etc && touch hosts.allow
terminal# mkdir run
terminal# chown -R  stunnel:stunnel run
```

В файл /usr/local/var/stunnel/hosts.allow пишем следующие строчки:

```
pop3s : ALL : allow
ALL : ALL : deny
```

В каталоге /usr/local/etc/stunnel/ создаем пару ключей stunnel.key (stunel.pem) ,stunel.csr, а также сертификат stunel.crt — это мы уже научились делать. Назначаем необходимые права на ключ. Если мы правильно установили popa3d (согласно документации по установке), то файл /etc/inetd.conf должен содержать строчку:

```
pop3    stream  tcp     nowait  root
/usr/local/sbin/popa3d popa3d
```

Все что нам остается — это перезагрузить демон inetd со следующими параметрами: -wWa 127.0.0.1, далее запускаем демон Stunnel:

```
terminal# /usr/local/sbin/stunnel
\ usr/local/etc/stunnel/stunnel.conf
```

Если в системе порт 995 не «забиндился» — значит, что-то было упущено, и нужно зачитать логи stunnel. Если все заработало, то можно проверять с помощью s_client наличие предоставляемого сертификата сервисом:

```
terminal# openssl s_client -host
localhost -port 995
```

Вкратце объясним, как работает данный механизм. Дело в том, что демон inetd делает службу popa3d доступной только из локального адреса, в свою очередь, демон stunel подключается к службе pop3 и становится посредником между клиентом и pop3-службой, организуя шифрованное соединение между собой и клиентом на своем порту (в данном случае — 995). Все просто и тривиально.

➜ **пишите письма!** Вот мы и подошли к логическому завершению. В результате мы, кажется, разобрались с основными понятиями криптографии с помощью открытого ключа, познакомились с такими понятиями, как ЭЦП и сертификат, научились теоретически и практически применять эти знания на практике в виде реальных рабочих систем. Удачи на ниве системного администрирования **C**

MAXI tuning
RUSSIAN EDITION
WWW.MAXI-TUNING.RU

ОН ТОЛЬКО ЧТО ПРОЧЕЛ MAXI tuning

В ПРОДАЖЕ СО 2 АВГУСТА

зловещий наблюдатель

МОНИТОРИНГ ПРОИЗВОДИТЕЛЬНОСТИ И РАБОТОСПОСОБНОСТИ BSD

ЧТО ДЕЛАТЬ, ЕСЛИ В ОДИН ЗЛОСЧАСТНЫЙ МОМЕНТ СИСТЕМА ПОЧТИ ОСТАНАВЛИВАЕТСЯ, ДИСК НАЧИНАЕТ БЕШЕНО ТРЕЩАТЬ, И РАБОТА ДАЖЕ В КОНСОЛИ СТАНОВИТСЯ НЕВЫНОСИМОЙ? КАК ВЫЯВИТЬ ПРОБЛЕМЫ И ВДОХНУТЬ В ОС ЖИЗНЬ? КАКИЕ МЕРЫ НЕОБХОДИМО ПРЕДПРИНЯТЬ, ЧТОБЫ ПОДОБНОЕ НЕ ПРОИЗОШЛО В БУДУЩЕМ? ОТВЕТЫ НА ЭТИ ВОПРОСЫ ТЫ НАЙДЕШЬ НИЖЕ

ЕВГЕНИЙ ЗОБНИН АКА J1M
{j1m@list.ru}

В поставку BSD-систем входит несколько программ, позволяющих провести анализ работоспособности отдельных компонентов ядра и всей системы в целом. Большинство из них являются представителями семейства утилит *stat (vmstat, iostat, pstat, fstat, netstat, systat) и предназначены для сбора статистики, другие служат несколько иным целям, но также занимают почетное место в инструментальном наборе системного администратора. И те, и другие являются незаменимыми помощниками системного администратора (и часто — пользователя домашнего ПК).

→ **выявляем обжор.** Традиционно, в UNIX-системах для выявления средней загруженности системы используют команду /usr/bin/uptime:

```
$ uptime
14:05:06 up 41 min, 4 users, load
average: 0.03, 0.08, 0.03
```

В выводе этой команды присутствует информация о текущем времени, времени, пройденном с момента загрузки системы, числе зарегистрированных пользователей и средней загруженности системы. Причем уровень нагрузки представлен не в процентах, а в усредненном количестве процессов, ждущих своего исполнения за последние 5, 10 и 15 минут. В слабо загруженной системе эти значения не поднимаются выше единицы. Внимательно следя за этими значениями, можно выявить момент, когда нагрузка на систему станет чрезмерной, и предпринять соответствующие меры. Значения нагрузки 5.0-6.0 говорят о том, что система загружена на 100%, и пора искать виновников проблемы.

Для выявления подозрительного процесса можно использовать команду /bin/ps с флагами '-aux'. В колонке %CPU для каждого процесса будет указано время использования процессора в про-

центах. Таким способом достаточно легко выявить проблемный процесс. Недостаток данного подхода состоит в том, что с его помощью можно увидеть только моментальный снимок состояния процессов. А для получения полной картины происходящего в реальном времени лучше использовать программу /usr/bin/top. Top, наверное, чаще всех остальных программ используется для выявления проблем с нагрузкой на процессор и память. Она выводит на экран список самых «прожорливых» процессов и регулярного его обновляет. Используя top, легко определить, какие из процессов наиболее требовательны к ресурсам.

Окно top разделено на две зоны: в верхней отображается различная системная статистика, в нижней — таблица «особо требовательных» процессов. Рассмотрим эти зоны подробнее. В первой строке отображается различная информация: последний PID, назначенный процессу, загруженность системы, время, прошедшее с момента загрузки и системное время. Вторая строка содержит информацию о процессах: общее количество и количество процессов, находящихся в каждом из пяти состояний: спячка (sleeping), исполнение (running), готов к запуску (starting), зомби (zombies), приостановлен (stopped). Третья строка сообщает нам о времени (в процентах), затрачиваемом процессором на выполнение кода программ (user), выполнение кода программ с повышенным приоритетом (nice), обработку системных вызовов (system), обработку прерываний (interrupt) и время простоя (idle). Четвертая строка предназначена для отображения информации о памяти: объем используемых станиц памяти (Active), объем неиспользуемых страниц памяти (то есть тех страниц, которые могут быть выгружены на диск, Inact), объем выгруженных страниц (необязательно в своп, — нужная страница может находиться в исполняемом файле, Wired), объем страниц, используемых для хранения дискового кэша (Cache), объем страниц, содержащих буферы (Buf) и объем свободных страниц. Пятая строка сообщает об объеме свопобласти и степени его заполненности.

Нижняя часть экрана отведена под таблицу процессов. Поля таблицы отображают следующую информацию: PID процесса (PID), имя владельца процесса (USERNAME), приоритет процесса, назначенный ядром (PRI), приоритет процесса, назначенный пользователем (NICE), общий размер процесса (SIZE), размер процесса в оперативной памяти (RES), состояние процесса (STATE), количество секунд процессорного времени, ушедшее на исполнение процесса (TIME), прогнозируемая загрузка процессора в процентах (WCPU), загрузка процессора (CPU).

Процесс, наиболее интенсивно использующий процессор, будет отображаться в первой строке. Самые «прожорливые» процессы обычно имеют высокие значения в колонках TIME и CPU. Не стоит доверять значениям колонки WCPU, они используются планировщиком задач и могут показывать совершенно дикие числа вроде 1000%.

Поведением top можно управлять с помощью интерактивных команд. Например, после нажатия Ctrl+L информация на экране немедленно обновиться, команда i убирает с экрана все спящие процессы, и на экране остается только информация о процессе, выполняющимся в данный момент. Отличный способ быстрого выявления негодяев. Существует еще несколько других команд, многие из которых имеют аналог в виде флага командной строки. Рассмотрим эти флаги:

опции командной строки top

-S — ПОКАЗЫВАТЬ СИСТЕМНЫЕ ПРОЦЕССЫ (КОМАНДА S)

-I — НЕ ПОКАЗЫВАТЬ СПЯЩИЕ ПРОЦЕССЫ (КОМАНДА I)

-T — НЕ ПОКАЗЫВАТЬ САМОГО СЕБЯ (КОМАНДА T)

```
last pid: 12969;  load averages:  1.13,  1.06,  0.67    up 0+00:51:38  08:35:14
28 processes:  2 running, 26 sleeping
CPU states: 15.7% user,  0.0% nice, 84.3% system,  0.0% interrupt,  0.0% idle
Mem: 16M Active, 61M Inact, 28M Wired, 5452K Cache, 22M Buf, 14M Free
Swap: 256M Total, 256M Free
  PID USERNAME PRI NICE   SIZE    RES STATE    TIME   WCPU    CPU COMMAND
11950 root       8    0  1656K  1164K wait     0:01  1.56%  1.42% sh
  473 jim       28    0  2984K  2432K pause    0:03  0.00%  0.00% zsh
  379 root      96    0  3476K  1988K select   0:01  0.00%  0.00% sendmail
11945 root       8    0  4280K  4176K wait     0:00  0.00%  0.00% make
11939 jim       96    0  2288K  1556K RUN      0:00  0.00%  0.00% top
  457 root       8    0  1632K  1128K wait     0:00  0.00%  0.00% login
  729 root      28    0  2540K  1924K pause    0:00  0.00%  0.00% zsh
  728 root       8    0  1620K  1120K wait     0:00  0.00%  0.00% login
  251 root      96    0  1324K   800K select   0:00  0.00%  0.00% syslogd
  395 root       8    0  1364K   916K nanslp   0:00  0.00%  0.00% cron
11919 root       8    0   804K   676K wait     0:00  0.00%  0.00% make
11912 root       8    0   544K   432K wait     0:00  0.00%  0.00% make
12969 root     121    0   524K   404K RUN      0:00  0.00%  0.00% make
11949 root       8    0   676K   560K wait     0:00  0.00%  0.00% make
  464 root       5    0  1288K   812K ttyin    0:00  0.00%  0.00% getty
  460 root       5    0  1288K   812K ttyin    0:00  0.00%  0.00% getty
11917 root       8    0  1648K  1152K wait     0:00  0.00%  0.00% sh
  459 root       5    0  1288K   812K ttyin    0:00  0.00%  0.00% getty
```

Внешний вид программы top

терминология vm

ACTIVE — ИСПОЛЬЗУЕМАЯ СТРАНИЦА ПАМЯТИ
INACTIVE — ЗАНЯТАЯ, НО НЕИСПОЛЬЗУЕМАЯ СТРАНИЦА (МОЖЕТ БЫТЬ ВЫГРУЖЕНА В SWAP-ОБЛАСТЬ)
WIRED OUT — ВЫГРУЖЕННАЯ СТРАНИЦА ПАМЯТИ
CACHE — СТРАНИЦА ИСПОЛЬЗУЕТСЯ ДЛЯ ХРАНЕНИЯ ДИСКОВОГО КЭША
BUF — СТРАНИЦА ИСПОЛЬЗУЕТСЯ ДЛЯ ХРАНЕНИЯ БУФЕРА ВВОДА/ВЫВОДА
FREE — СВОБОДНАЯ СТРАНИЦА

-MCPU — СТАТИСТИКА CPU

-MIO — СТАТИСТИКА ВВОДА/ВЫВОДА (КОМАНДА M)

-Q — УСТАНОВИТЬ ПРОЦЕССУ TOP НАИВЫСШИЙ ПРИОРИТЕТ

-U — ПОКАЗЫВАТЬ UID ВМЕСТО ИМЕНИ ПОЛЬЗОВАТЕЛЯ

-S — ВРЕМЕННОЙ ПРОМЕЖУТОК МЕЖДУ ОБНОВЛЕНИЯМИ (КОМАНДА S, ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ — 2)

-U — ПОКАЗЫВАТЬ ПРОЦЕССЫ ТОЛЬКО ДАННОГО ЮЗЕРА (КОМАНДА U)

-I — ПОКАЗЫВАТЬ НИТИ (КОМАНДА H)

Флаги '-q' и '-u' очень полезны в тех случаях, когда нагрузка на систему столь велика, что top не в состоянии быстро загрузиться.

➔ **статистика виртуальной памяти.** Если треск жесткого диска стал очень частым и надоедливым, то это свидетельствует об одном — о высокой интенсивности операций подкачки. При острой нехватке оперативной памяти подсистема VM ядра начинает лихорадочно работать со swap-областью, выгружая и вновь загружая не умещающиеся в основной памяти данные. Чтобы убедиться в правоте этого высказывания, достаточно выполнить команду /usr/sbin/swapinfo или /usr/sbin/pstat -s и посмотреть на степень заполненности swap-области.

Для получения подробной статистики о работе подсистемы виртуальной памяти обычно используют команду /usr/bin/vmstat. Ее можно найти практически в любой UNIX-системе, начиная с HP-UX и заканчивая Linux. Vmstat печатает информацию, разбивая ее на шесть тематических разделов: procs — информация о процессах, memory — количество доступной памяти, page — активность страничной подкачки, disks — операции с диском, faults — переключения контекста и прерывания, cpu — использование процессора. Не вся эта информация связана напрямую с подсистемой виртуальной памяти, но печатается для того, чтобы можно было по-

лучить полную картину происходящего и быстро определить причины возникновения той или иной ситуации. Каждый из перечисленных разделов содержит несколько колонок. Разберем их подробнее:

статистика vmstat

PROCS:

R — КОЛИЧЕСТВО ГОТОВЫХ К ЗАПУСКУ ПРОЦЕССОВ

B — КОЛИЧЕСТВО ЗАБЛОКИРОВАННЫХ В ОЖИДАНИИ РЕСУРСОВ ПРОЦЕССОВ

W — КОЛИЧЕСТВО ГОТОВЫХ К ЗАПУСКУ, НО ПЕРЕМЕЩЕННЫХ В СВОП, ПРОЦЕССОВ

MEMORY:

AVM — ОБЪЕМ ДОСТУПНОЙ ВИРТУАЛЬНОЙ ПАМЯТИ

FRE — ОБЪЕМ СВОБОДНОЙ ПАМЯТИ

PAGE:

FLT — КОЛИЧЕСТВО «ПРОМАХОВ» (ОБРАЩЕНИЙ К СТРАНИЦАМ, КОТОРЫХ НЕТ В ДАННЫЙ МОМЕНТ В ОПЕРАТИВНОЙ ПАМЯТИ)

RE — ЧИСЛО ВОЗВРАЩЕННЫХ (ВОССТАНОВЛЕННЫХ ИЗ СПИСКА НЕИСПОЛЬЗУЕМЫХ) СТРАНИЦ

PI — КОЛИЧЕСТВО ПОДКАЧЕННЫХ СТРАНИЦ

PO — КОЛИЧЕСТВО ВЫГРУЖЕННЫХ СТРАНИЦ

FR — КОЛИЧЕСТВО ОСВОБОЖДЕННЫХ СТРАНИЦ

SR — ЧИСЛО СТРАНИЦ, ОБРАБОТАННЫХ ПО АЛГОРИТМУ «ЧАСЫ» (АЛГОРИТМ «ЧАСЫ» ИСПОЛЬЗУЕТСЯ ДЛЯ ПОМЕТКИ ДАВНО НЕ ИСПОЛЬЗОВАВШИХСЯ СТРАНИЦ)

FAULTS:

IN — ЧИСЛО ПРЕРЫВАНИЙ

SY — ЧИСЛО СИСТЕМНЫХ ВЫЗОВОВ

CS — ЧИСЛО ПЕРЕКЛЮЧЕНИЙ КОНТЕКСТА

DISKS:

ЧИСЛО ОПЕРАЦИЙ С ДИСКАМИ

CPU:

US — ВРЕМЯ, ИСТРАЧЕННОЕ НА ИСПОЛНЕНИЕ КОДА ПРОГРАММ

SY — ВРЕМЯ, ИСТРАЧЕННОЕ НА ИСПОЛНЕНИЯ КОДА ЯДРА (СИСТЕМНЫЕ ВЫЗОВЫ, ВВОД/ВЫВОД)

ID — ВРЕМЯ ПРОСТОЯ

Информация в разделах page, fault и disks представлена в форме «число в секунду», в остальных разделах значения приводятся на момент снятия информации. По умолчанию vmstat печатает данные один раз, и, чтобы увидеть статистику в реальном времени, следует запускать программу так: «vmstat <интервал> <количество повторов>». Понаблюдав некоторое время за листингом, можно определить, все ли в порядке, или что-то идет не так. Например, если значение в колонке w раздела procs часто становится больше нуля, значит, готовые к работе процессы перемещаются в своп, и это говорит о нехватке памяти. О том, что памяти не хватает, также можно узнать, понаблюдав за полем fr раздела page. Если значение этого поля становится слишком высоким, значит, либо завершилось исполнение большой программы, либо памяти просто не хватает. Зная принципы работы подсистемы виртуальной памяти и постоянно просматривая листинги vmstat, можно почти со стопроцентной вероятностью определить, в какой момент системе начнет не хватать памяти.

Полную статистику за все время непрерывной работы ОС можно узнать, указав флаг '-s'. В выводе будет присутствовать информация не только о количестве занятых и освобожденных страниц, но также общее число системных вызовов, вызовов fork(2) и другой статистической информации. Статистика количества прерываний, поступивших от устройств, выводится после указания флага '-i'. Любопытной для интересующихся внутренностями ядра может оказаться информация о памяти, выделенной ядром для своих нужд (флаг '-m').

➔ **статистика ввода/вывода.** Для наблюдения за пропускной способностью жесткого диска в BSD (да и в большинстве UNIX-систем) используют команду /usr/bin/iostat. Формат выходной информации этой команды сходен с форматом команды vmstat. Данные также разбиты на разделы. В первых двух колонках tin и tout (раздел tty) содержится информация о числе символов, введенных и выведенных терминалами и псевдотерминалами за секунду (практически бесполезная информация). Каждый из последующих разделов соответствует конкретному блочному устройству. За устройством закрепляется три колонки: KB/t — объем информации (в килобайтах), переданный за одну пересылку данных, tps — количество пересылок данных в секунду, MB/S — объем данных (в мегабайтах), переданных за секунду. В последнем разделе, по традиции, содержится статистика использования процессора. Iostat показывает столько разделов устройств, сколько умещается на дисплей 80 на 25 символов (причем независимо от реального размера терминала). К счастью, можно указать программе не показывать разделы tty и cpu, запустив ее с флагом '-d'. Просматривая листинги vmstat, можно определить, работает ли жесткий диск с полной скоростью или его что-то ограничивает (например, PIO-режим вместо DMA).

Запустив команду с флагом '-I', можно также просмотреть статистику за все время работы системы, а не только данные о количестве переданной информации в секунду. В этом случае к каждому устройству будут привязаны такие колонки: KB/t — объем информации (в килобайтах), переданный за одну пересылку данных, xfrs — общее количество пересылок данных, MB — общий объем переданных данных.

➔ **один за всех.** В стандартную поставку BSD-систем входит программа, аккумулирующая возможности всех рассмотренных выше программ. Имя ей systat, и она не только может представить нам практически все возможности таких программ, как vmstat, iostat и netstat, но и делает это в более понятной форме в реальном времени. Будучи основанной на библиотеке ncurses, systat не печатает данные последовательно на терминале, а создает собственное окно (окно не в терминологии X Window, а в терминологии ncurses) и регулярно обновляет его.

При запуске systat делит экран на две области. В верхней всегда отображается степень нагрузки на систему (та самая, о которой сообщает команда uptime), содержимое нижней зависит от режима, в котором работает программа. При запуске без флагов в нижней части отобразится лишь информация о процессе, наиболее интенсивно использующем процессор. Для смены режима работы следует перейти в командный режим (ввести ':' как в vi и less) и ввести его имя или указать имя в командной строке (не забыв про знак '-').

Всего существует 11 режимов: pigs — стандартный, swap — информация о swap-областях, iostat — статистика ввода/вывода, vmstat — статистика виртуальной памяти, netstat — сетевая статистика, icmp, ip, icmp6, ip6, tcp — количество переданных/посланных сообщений по определенному протоколу, ifstat — количество переданного трафика по каждому из сетевых интерфейсов **C**

top-like программы

XTOP — ВЕРСИЯ ПРОГРАММЫ ДЛЯ X-WINDOW
KTOP — TOP ДЛЯ KDE
HTOP — ДРУЖЕЛЮБНЫЙ TOP
NTOP — МОНИТОРИНГ СЕТЕВОЙ АКТИВНОСТИ
MYTOP & MTOP — МОНИТОРИНГ ЗАПРОСОВ К MYSQL
DNSTOP — МОНИТОРИНГ DNS-ЗАПРОСОВ
ITOP — МОНИТОРИНГ ГЕНЕРАЦИИ ПРЕРЫВАНИЙ
APACHETOP — МОНИТОРИНГ ПОПУЛЯРНОГО WEB-СЕРВЕРА


Липкая Липка
Мы немного намочили ведущую Муз-ТВ
Тушим свет!
Телевидение и домашнее кино завтрашнего дня
Автосекс
Лучшие места «парковки»
KaZантип 2006
Секретная карта предстоящего угара
40
горячих новинок лета
160 страниц
Техника как стиль жизни
SYNC
В продаже с 28 июня
ЖУРНАЛ СИНК

корень зла

WEB-СЕРВЕР В СРЕДЕ CHROOT: ПРАКТИЧЕСКАЯ ПАРАНОЙЯ

ЧТО ДЕЛАТЬ, ЕСЛИ В ОДИН ЗЛОСЧАСТНЫЙ МОМЕНТ СИСТЕМА ПОЧТИ ОСТАНАВЛИВАЕТСЯ, ДИСК НАЧИНАЕТ БЕШЕНО ТРЕЩАТЬ, И РАБОТА ДАЖЕ В КОНСОЛИ СТАНОВИТСЯ НЕВЫНОСИМОЙ? КАК ВЫЯВИТЬ ПРОБЛЕМЫ И ВДОХНУТЬ В ОС ЖИЗНЬ? КАКИЕ МЕРЫ НЕОБХОДИМО ПРЕДПРИНЯТЬ, ЧТОБЫ ПОДОБНОЕ НЕ ПРОИЗОШЛО В БУДУЩЕМ? ОТВЕТЫ НА ЭТИ ВОПРОСЫ ТЫ НАЙДЕШЬ НИЖЕ

ANDREY MATVEEV
{andrushock@real.xakep.ru}

Сегодня у нас на повестке дня вопросы, касающиеся обеспечения безопасности Web-сервера на базе OpenBSD. Мы научим PHP, MySQL и Sendmail работать с Apache, который запускается в окружении chroot — измененном корневом каталоге /var/www с правами непривилегированного пользователя www. Таким образом, нам удастся обеспечить дополнительный уровень защиты и максимально снизить возможный ущерб в том случае, если злоумышленнику окажется под силу взломать нашу систему. Применив полученные знания на практике, в твоем арсенале будет настолько защищенная система, что ты сможешь совершенно спокойно взяться за разработку любого проекта, будь то личный блог, новостной сайт компании или даже интернет-магазин с мириадами клиентов.

➜ **предварительный ликбез.** Последняя версия ультрасекьюрной OpenBSD (3.9 на момент написания статьи) как нельзя лучше подойдет для выполнения нашей миссии. Короткая история взломов, прекрасная реализация стека TCP/IP, отличный файрвол Packet Filter (pf), залатанный Apache 1.3.29 с поддержкой SSL, наличие последних версий OpenSSH и OpenSSL, тысячи добротно протестированных прекомпилированных пакетов — все это говорит в пользу сделанного выбора. Хотя стоит отметить, что в качестве используемой операционной системы может выступать любая из Free/Net/DragonFlyBSD.

За основу нашей конструкции примем PHP и MySQL. Не секрет, что за последние годы эта связка стала стандартом де-факто для интернет-проектов различного масштаба. Поэтому давай не будем на этом останавливаться и перейдем непосредственно к настройке.

➜ **каждой службе — свой раздел.** Прежде всего необходимо грамотно подойти к разделению дискового пространства. Лично я предпочитаю для каждой критически важной сетевой службы выделять собственный раздел. Для наглядности приведу содержимое конфига fstab(5) полностью:

```
# vi /etc/fstab
/dev/wd0a / ffs rw 1 1
/dev/wd1h /backup ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd1a /cvs ffs rw,nodev,nosuid 1 2
/dev/wd0g /export ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd1i /home ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd0d /tmp ffs rw,nodev,nosuid,
noexec,softdep 1 2
/dev/wd0f /usr ffs rw,nodev,softdep 1 2
/dev/wd0e /var ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd1d /var/mail ffs rw,nodev,
nosuid,noatime,softdep 1 2
/dev/wd1e /var/mysql ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd1f /var/squid ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd1g /var/www ffs rw,nodev,
nosuid,softdep 1 2
/dev/wd1b none swap sw 0 0
```

Преимущества такой конфигурации видны даже невооруженным глазом:

- ПЕРЕПОЛНЕНИЕ ОДНОГО ИЗ РАЗДЕЛОВ НЕ ПОВЛИЯЕТ НА РАБОТУ БОЛЬШИНСТВА СЛУЖБ;
- ПРИ СЛУЧАЙНОМ ОТКЛЮЧЕНИИ ПИТАНИЯ ВО ВРЕМЯ ВЫПОЛНЕНИЯ ОПЕРАЦИИ ЗАПИСИ НА ОДНУ ИЗ ФАЙЛОВЫХ СИСТЕМ СНИЖАЕТСЯ ВОЗМОЖНОСТЬ ПОВРЕЖДЕНИЯ ОСТАЛЬНЫХ ФС;
- УВЕЛИЧЕНИЕ БЕЗОПАСНОСТИ СИСТЕМЫ ЗА СЧЕТ УКАЗАНИЯ ДЛЯ КАЖДОЙ ФС СПЕЦИАЛЬНЫХ ФЛАГОВ МОНТИРОВАНИЯ.

→ **MySQL: записная книжка с SQL-интерфейсом.** Теперь перейдем к установке и конфигурированию сервера баз данных. Собираем из портов три MySQL-пакета: client, server, tests:

```
# cd /usr/ports/databases/mysql
# make package
```

Для проверки правильности сборки выполняем набор прилагаемых тестов:

```
# make do-regress
```

Если все тесты прошли успешно, переходим к установке прекомпилированного пакета клиентской части MySQL:

```
# pkg_add /usr/ports/packages/-
i386/all/mysql-client-5.0.22.tgz
```

Уделяем внимание зависимостям mysql-server'а:

```
# cd /usr/ports/databases/p5-DBD-mysql
# make install clean CLEANDEPENDS=Yes
```

И устанавливаем пакадж серверной части MySQL:

```
# pkg_add /usr/ports/packages/-
i386/all/mysql-server-5.0.22.tgz
```

При необходимости создаем директорию /var/mysql:

```
# mkdir -p /var/mysql
# chown _mysql:_mysql /var/mysql
```

Настало время проверить работоспособность БД и запустить скрипт для создания типовых баз mysql и test:

```
# /usr/local/bin/mysql_install_db
```

Теперь в стартовом сценарии /etc/rc.local указываем опции для mysqld_safe — своего рода обертки, которая запускает mysqld с заданными параметрами, производит мониторинг состояния демона и при необходимости перезапускает главный процесс MySQL.

 # vi /etc/rc.local

```
if [ -x /usr/local/bin/mysqld_safe ]; then
  echo -n ' mysqld'
  /usr/local/bin/mysqld_safe --user=_mysql \
    --open-files=1000 --skip-networking \
    --socket=/var/www/var/run/mysql/-
mysql.sock &
fi
```

Прокомментирую опции, которые мы указали при старте MySQL:

— USER=_MYSQL — ЗАПУСК ДЕМОНА ОТ ИМЕНИ НЕПРИВИЛЕГИРОВАННОГО ПОЛЬЗОВАТЕЛЯ _MYSQL;

— OPEN-FILES=1000 — МАКСИМАЛЬНОЕ ЧИСЛО ОТКРЫТЫХ ФАЙЛОВ;

— SKIP-NETWORKING — MYSQLD НЕ ДОЛЖЕН БИНДИТЬСЯ НА СЕТЕВЫЕ АДРЕСА;

— SOCKET=/VAR/WWW/VAR/RUN/MYSQL/MYSQL.SOCK — ТАК КАК НАША БД БУДЕТ ИСПОЛЬЗОВАТЬСЯ ТОЛЬКО ЛОКАЛЬНО УСТАНОВЛЕННЫМИ ПРОГРАММАМИ, РАБОТАЕМ ЧЕРЕЗ СОКЕТ;

— '&' — ВЫПОЛНЯЕМ ПЕРЕВОД MYSQLD_SAFE В ФОНОВЫЙ РЕЖИМ.

Чтобы не изобретать велосипед, давай возьмем предлагаемый разработчиками пример конфига MySQL, назначим ему корректные права доступа и отредактируем применительно к нашим задачам:

```
# cp /usr/local/share/mysql/my-medi-
um.cnf /etc/my.cnf
# chmod 644 /etc/my.cnf
```

 # vi /etc/my.cnf

```
[client]
socket = /var/www/var/run/mysql/mysql.sock
[mysqld]
socket = /var/www/var/run/mysql/mysql.sock
skip-locking
key_buffer = 16M
max_allowed_packet = 1M
table_cache = 64
sort_buffer_size = 512K
net_buffer_length = 8K
myisam_sort_buffer_size = 8M
[mysqldump]
quick
max_allowed_packet = 16M
[mysql]
no-auto-rehash
```

Как ты мог заметить, главное отличие от дефолтного my-medium.cnf заключается в определении местоположения абсолютного пути до сокета клиента и сервера MySQL. Вместо /var/run/mysql/mysql.sock мы будем использовать /var/www/var/run/mysql/mysql.sock, поэтому своевременно подготавливаем соответствующую поддиректорию:

```
# mkdir -p /var/www/var/run/mysql
# chown _mysql:_mysql /var/www/var/run/mysql
```

Для хранения временных файлов необходимо создать каталог /var/www/tmp с либеральными правами доступа:

```
# mkdir -p -m 777 /var/www/tmp
```

С установкой и конфигурированием разобрались, переходим к запуску демона на орбиту:

```
# /usr/local/bin/mysqld_safe
--user=_mysql --open-files=1000 \
--skip-networking --socket=/var/www/-
var/run/mysql/mysql.sock &
```

Проверяем, готов ли mysqld принимать входящие подключения:

```
# fstat | grep mysql
_mysql   mysqld   22841   wd /var/mysql
         2 drwxr-xr-x    r        1024
_mysql   mysqld   22841   0 /      67197
crw-rw-rw-        r       null
_mysql   mysqld   22841   1 /var/mysql
         3 -rw-rw----    w        25648
_mysql   mysqld   22841   2 /var/mysql
         3 -rw-rw----    w        25648
_mysql   mysqld   22841   3 pipe
0xd69e8828 state
```

И создаем (для надежности) символическую ссылку на стандартное расположение сокета:

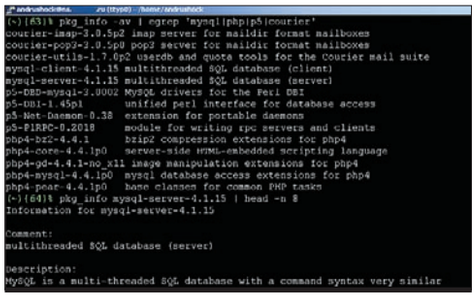
```
# ln -sf /var/www/var/run/my-
sql/mysql.sock /var/run/mysql/mysql.sock
```

→ **hardening MySQL в подробностях.** Опциональный шаг: выполняем ряд несложных операций по увеличению безопасности MySQL:

 # /usr/local/bin/mysql -u root

```
// Пустой пароль для администратора
MySQL-сервера нам не подходит,
устанавливаем новый
mysql> set password for
root@localhost=password("noidea");
// Удаляем базу данных test, которая
была создана скриптом mysql_install_db
mysql> drop database test;
// Удаляем все MySQL'ные учетные
записи, кроме root
mysql> use mysql;
mysql> delete from db;
mysql> delete from user where not
host="localhost" and user="root");
mysql> flush privileges;
// Чтобы усложнить атаки типа
bruteforce, можно изменить
имя главной учетной записи
с root на admin
mysql> update user set user="admin"
where user="root";
mysql> flush privileges;
// Настройка закончена
mysql> quit
```

→ **хардкорные разборки с PHP.** Далее на очереди — PHP4 со своими расширениями. Следующими командами мы установим пакет с основным движком — так называемый core-пакадж, модуль для работы с базами данных и библиоте-



Список установленных прекомпилированных пакетов

ку pear (набор специальных компонентов и расширений для PHP).

```
# pkg_add php4-core-4.4.1p0.tgz
# pkg_add php4-mysql-4.4.1p0.tgz
# pkg_add php4-pear-4.4.1p0.tgz
```

Активируем модуль libphp4.so:

```
# /usr/local/sbin/phpxs -s
```

Воспользуемся рекомендованной разработчиками версией php.ini:

```
# cp /usr/local/share/examples/php4/-
php.ini-recommended /var/www/conf/php.ini
```

Выставляем корректные права доступа:

```
# chown root:www /var/www/conf/php.ini
# chmod 640 /var/www/conf/php.ini
```

Указываем путь до сокета MySQL:

 # vi /var/www/conf/php.ini

```
mysql.default_socket = /var/run/my-
sql/mysql.sock
```

И активируем MySQL-модуль:

```
# /usr/local/sbin/phpxs -a mysql
```

→ **сам себе надежный почтальон.** Для корректной работы электронной почты в Apache chroot необходимо установить статически слинкованную версию mini_sendmail. Этот фэйковый почтовик будет передавать из среды chroot всю исходящую почту полноценному транспортному агенту.

```
# cd /usr/port/mail/mini_sendmail
# env SUBPACKAGE=-chroot make install
# cp -p /bin/sh /var/www/bin
# mkdir -p /var/www/etc
# cp /etc/{hosts,resolv.conf} /var/www/etc
```

Теперь снова возвращаемся в php.ini и указываем абсолютный путь до mini_sendmail относительно /var/www (внимание: запись «-fwww@mydomain.ru» обязательно должна идти без пробела).

 # vi /var/www/conf/php.ini

```
sendmail_path = "/bin/mini_sendmail -
fwww@mydomain.ru -t"
```

→ **не выпускай суккуба из песочницы.** Разработчики OpenBSD выполнили львиную долю работы за нас, посадив Apache/mod_ssl в chroot-окружение. Нам лишь остается активировать поддержку PHP и разобраться с виртуальными хостами. Для этого переходим к редактированию главного конфигурационного файла индейца:

 # vi /var/www/conf/httpd.conf

```
// Подгружаем модуль PHP4
LoadModule php4_module
/usr/lib/apache/modules/libphp4.so
// Добавляем PHP'шные классы к типу MIME
<IfModule mod_php4.c>
AddType application/x-httpd-php .php .php4
AddType application/x-httpd-php-source .phps
</IfModule>
// Расширяем список файлов, которые при
WWW-запросе будут обрабатываться
в первую очередь
DirectoryIndex index.html index.php
index.php4
// Указываем виртуальные интерфейсы
в данном случае используется
виртуальный хостинг на основе имен)
NameVirtualHost 192.168.1.1
NameVirtualHost 192.168.3.1
NameVirtualHost 212.XX.XY.162
// Внутри контейнера VirtualHost задаем
параметры конфигурации для www.mydomain.ru
<VirtualHost
192.168.1.1 192.168.3.1 212.XX.XY.162>
 ServerAdmin admin@mydomain.ru
 DocumentRoot /var/www/virtual/
 www.mydomain.ru
 ServerName www.mydomain.ru
 ServerAlias mydomain.ru
 ErrorLog logs/virtual.www.mydo-
 main.ru-error_log
 CustomLog logs/virtual.www.mydo-
 main.ru-access_log common
</VirtualHost>
// Определяем списки контроля доступа
для директории с файлами, предназначенными
только для администрирования CMS
<Directory "/var/www/virtual/www.mydo-
main.ru/admin">
 Order deny,allow
 Deny from all
 Allow from localhost
192.168.1.0/24 192.168.3.0/24 212.XX.XY.162
</Directory>
```

→ **управляем MySQL с комфортом.** PHPMyAdmin представляет собой набор PHP-скриптов для управления сервером MySQL. Прекрасно подходит для поклонников визуального администрирования и тех, у кого синтаксис SQL-запросов вызывает определенную сложность. Кроме того, с помощью PHPMyAdmin довольно удобно выполнять рутинные операции по бэкапу, созданию и модификации баз данных, таблиц, пользователей и т.д. Устанавливаем:

```
# ftp http://switch.dl.sourceforge.net/-
sourceforge/phpmyadmin/-
phpMyAdmin-2.8.2.tar.gz
# tar zxvf phpMyAdmin-2.8.2.tar.gz
# mkdir -p /var/www/virtual
# cp -Rp phpMyAdmin-2.8.2 /var/www/-
virtual/phpmyadmin.mydomain.ru
# cd /var/www/virtual/-
phpmyadmin.mydomain.ru
# cp libraries/config.default.php
config.inc.php
```

В конфиге config.inc.php указываем, что в качестве типа соединения у нас используется «сокет» (напомню, mysqld не подвешен даже на интерфейс обратной петли), а также имя и пароль администратора MySQL:

 # vi config.inc.php

```
$cfg['Servers'][$i]['socket']    = '';
$cfg['Servers'][$i]
['connect_type']             = 'socket';
$cfg['Servers'][$i]
['auth_type']                = 'config';
$cfg['Servers'][$i]['user'] = 'admin';
$cfg['Servers'][$i]
['password']                 = 'noidea';
```

Описание поддомена phpmyadmin.mydomain.ru в httpd.conf будет выглядеть следующим образом:

 # vi /var/www/conf/httpd.conf

```
<VirtualHost 192.168.1.1 192.168.3.1>
 ServerAdmin admin@mydomain.ru
 DocumentRoot /var/www/virtual/phpmyad-
 min.mydomain.ru
 ServerName phpmyadmin.mydomain.ru
 ErrorLog logs/virtual.phpmyadmin.mydo-
 main-error_log
 CustomLog logs/virtual.phpmyadmin.my-
 domain-access_log common
</VirtualHost>
```

Совершенно очевидно, что доступ к phpMyAdmin необходимо ограничить. Этого можно добиться разными способами. Для расширения кругозора предлагаю воспользоваться аутентификацией по паролю. Чтобы проконтролировать доступ к каталогу /var/www/virtual/phpmyadmin.mydomain.ru и запретить по сети передавать пароли в открытом виде (директива SSLRequireSSL), создаем еще один управляющий файл — .htaccess. Преимущество использования такого подхода состоит в том, что мы не захламляем httpd.conf дополнительными ди-

рективами для описания правил доступа, указания местонахождения Auth-конфигов и методов аутентификации. Плюс к этому, при изменении конфигурации в файле .htaccess не придется перезагружать Web-сервер.

 # vi /var/www/virtual/phpmyadmin.mydomain.ru /.htaccess

```
SSLRequireSSL
AuthType Basic
AuthName "Password Required"
AuthUserFile /var/www/conf/.htpasswd
AuthGroupFile /dev/null
<Limit GET POST>
require user admin
</Limit>
```

Аутентификационную базу /var/www/conf/.htpasswd (ни в коем случае не размещай .htpasswd в каталоге /var/www/virtual/phpmyadmin.mydomain.ru) будем вести с помощью утилиты htpasswd(1). Ключ '-c' отвечает за создание базы, ключ '-m' задает использование алгоритма шифрования MD5 вместо применяемой по умолчанию DES'овской функции crypt(3):

```
# htpasswd -cm /var/www/conf/.htpasswd admin
```

Только суперпользователь и демон httpd имеют право обращаться к базе с паролями:

```
# chown root:www /var/www/conf/.htpasswd
# chmod 640 /var/www/conf/.htpasswd
```

➔ **проводим безопасные транзакции по протоколу https.** Чтобы получить возможность устанавливать защищенные сеансы по протоколу https, необходимо создать приватный ключ, ввести регистрационные данные и подписать сертификат собственным ключом. Начнем с генерации секретного RSA-ключа длиной 1024 бит:

```
# openssl genrsa -out /etc/ssl/private/-
server.key 1024
```

Создаем запрос на сертификат:

 # openssl req -new -key /etc/ssl/private/server.key \ -out /etc/ssl/private/server.csr

```
Country Name (2 letter code) []:RU
State or Province Name (full name) []:Russia
Locality Name (eg, city) []:Moscow
Organization Name (eg, company) []:MySite
Organizational Unit Name (eg, section)
[]:<Enter>
Common Name (eg, fully qualified host
name) []:www.mydomain.ru
Email Address []:admin@mydomain.ru
A challenge password []:<Enter>
An optional company name []:<Enter>
```

Подписываем сертификат, который будет действовать на протяжении 10 лет (аргумент '-days 3650'), своим ключом:

```
# openssl x509 -req -days 3650 -in
/etc/ssl/private/server.csr \
 -signkey /etc/ssl/private/server.key-
out /etc/ssl/server.crt
```

Выполняем остановку и повторный запуск httpd(8), но уже с поддержкой https:

```
# apachectl stop
# apachectl startssl
```

Проверяем, забиндился ли апач на соответствующие порты:

```
% netstat -na -f inet | egrep '80|443'
tcp 0 0 *.80      *.*      LISTEN
tcp 0 0 *.443     *.*      LISTEN
```

В /etc/pf.conf создаем правило, разрешающее прохождение запросов к портам 80 и 443:

 # vi /etc/pf.conf

```
$ext_if = "fxp0"
pass in log on $ext_if inet proto tcp
from any to $ext_if \
   port { www, https } flags S/SA keep state
```

Чтобы внесенные изменения непременно вступили в силу не забудь перезагрузить набор рулесетов файрвола:

```
# pfctl -f /etc/pf.conf
```

В конфиге /etc/rc.conf следующими записями разрешаем автоматическую загрузку Apache SSL и Packet Filter при старте системы:

 # vi /etc/rc.conf

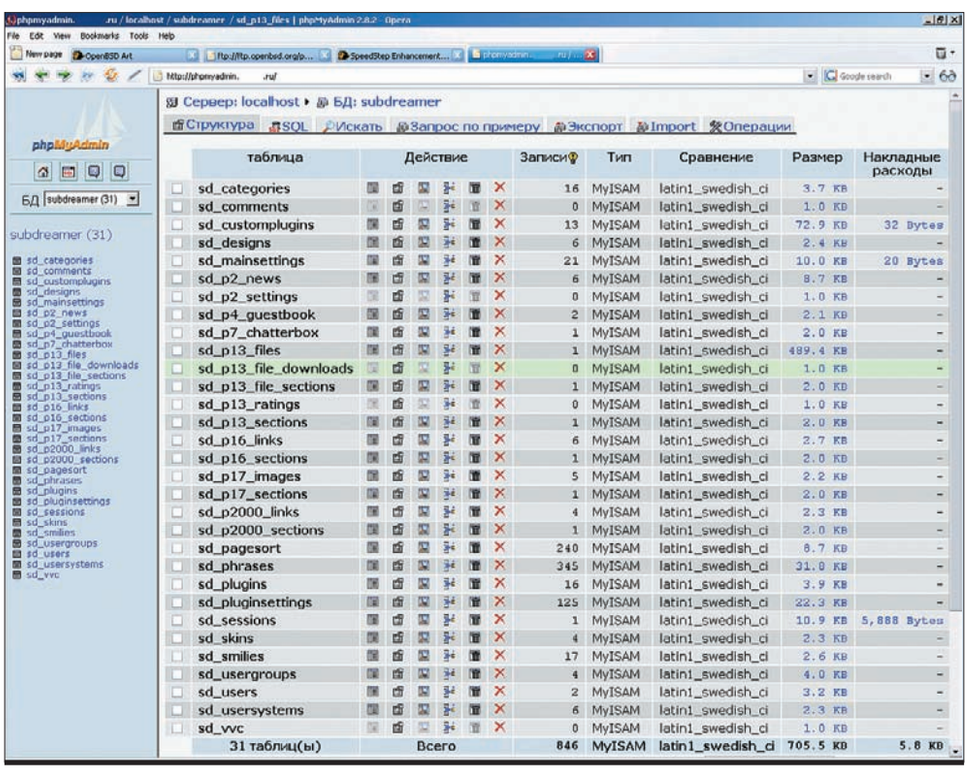
```
httpd_flags="-DSSL"
pf=YES
```

➔ **постскриптумы.** На этом настройку можно считать завершенной. Протестировать работу модуля php4 и его взаимодействие с MySQL можно с помощью php-файла, исходный код которого приведен ниже.

 # vi /var/www/virtual/www.mydomain.ru/test.php

```
<html><body>
<?php
mysql_connect("localhost", "admin",
"noidea") or die("failed");
print "ok";
mysql_close();
?>
</body></html>
```

Если при запросе **www.mydomain.ru/test.php** ты получишь лаконичный ответ «ok» — смело начинай заливать любимую CMS'ку в каталог /var/www/virtual/www.mydomain.ru. Удачи C



Администрируем MySQL через Web-интерфейс

восставшие из ада

ВОССТАНОВЛЕНИЕ УДАЛЕННЫХ ФАЙЛОВ

КТО НЕ УДАЛЯЛ ФАЙЛОВ И ПОТОМ ГОТОВ БЫЛ ПОВЕСИТЬСЯ, ЧТОБЫ ВЕРНУТЬ ИХ ОБРАТНО? ОСОБЕННО ЛЕГКО УДАЛЯТЬ ДАННЫЕ С ПОМОЩЬ КОМАНДНОЙ СТРОКИ, КОГДА ЛИШНИЙ ПРОБЕЛ ИЛИ СИМВОЛ ЗВЕЗДОЧКИ ТРУТ ВСЕ ПОДЧИСТУЮ. ХОЧЕШЬ УЗНАТЬ, КАК ЭТОМУ ПРОТИВОСТОЯТЬ?

КРИС КАСПЕРСКИ АКА МЫЩЪХ

xBSD поддерживает множество файловых систем: FAT16/32, ext2fs/ext3fs, ISO 9660, UDF, NFS, SMBFS, NTFS, ReiserFS, XF, AFS, LFS, но основной системой, устанавливаемой по умолчанию, была всегда и остается UFS/UFS2.

Многие коммерческие UNIX'ы также используют либо саму UFS, либо нечто очень на нее похожее. В противоположность ext2fs, исхоженной вдоль и поперек, UFS (равно как и ее наследница FFS) практически недокументированна и в доступной литературе описана поверхностно. Единственным источником информации становятся исходные тексты, в которых не так-то просто разобраться!

Существует множество утилит, восстанавливающих уничтоженные данные (или, во всяком случае, пытающихся делать это), но на проверку все они оказываются неработоспособными (или обнаруживают не все файлы), что, в общем-то, и неудивительно, поскольку автоматическое восстановление удаленных файлов под UFS невозможно в принципе. Тем не менее, это достаточно легко сделать вручную, если, конечно, знать как.

→ **что происходит при удалении файла.** При удалении файла на UFS-разделе происходит следующее (события перечислены в порядке расположения соответствующих структур в разделе и могут не совпадать с порядком их возникновения):

[1] В СУПЕРБЛОКЕ ОБНОВЛЯЕТСЯ ПОЛЕ FS_TIME (ВРЕМЯ ПОСЛЕДНЕГО ДОСТУПА К РАЗДЕЛУ).

[2] В СУПЕРБЛОКЕ ОБНОВЛЯЕТСЯ СТРУКТУРА FS_CSTOTAL (КОЛИЧЕСТВО СВОБОДНЫХ INODE И БЛОКОВ ДАННЫХ).

[3] В ГРУППЕ ЦИЛИНДРОВ ОБНОВЛЯЮТСЯ КАРТЫ ЗАНЯТЫХ INODE И БЛОКОВ ДАННЫХ. INODE И ВСЕ БЛОКИ ДАННЫХ УДАЛЯЕМОГО ФАЙЛА ПОМЕЧАЮТСЯ КАК ОСВОБОЖДЕННЫЕ.

[4] В INODE МАТЕРИНСКОГО КАТАЛОГА ОБНОВЛЯЮТСЯ ПОЛЯ ВРЕМЕНИ ПОСЛЕДНЕГО ДОСТУПА И МОДИФИКАЦИИ.

[5] В INODE МАТЕРИНСКОГО КАТАЛОГА ОБНОВЛЯЕТСЯ ПОЛЕ ВРЕМЕНИ ПОСЛЕДНЕГО ИЗМЕНЕНИЯ INODE.

[6] В INODE УДАЛЯЕМОГО ФАЙЛА ПОЛЯ DI_MODE (IFMT, PERMISSIONS), DI_NLINK (КОЛИЧЕСТВО ССЫЛОК НА ФАЙЛ) И DI_SIZE (РАЗМЕР ФАЙЛА) ВАРВАРСКИ ОБНУЛЯЮТСЯ.

[7] В INODE УДАЛЯЕМОГО ФАЙЛА ПОЛЯ DI_DB (МАССИВ УКАЗАТЕЛЕЙ НА 12 ПЕРВЫХ БЛОКОВ ФАЙЛА) И DI_IB (УКАЗАТЕЛЬ НА БЛОК КОСВЕННОЙ АДРЕСАЦИИ) БЕЗЖАЛОСТНО ЗАТИРАЮТСЯ НУЛЯМИ.

[8] В INODE УДАЛЯЕМОГО ФАЙЛА ОБНОВЛЯЮТСЯ ПОЛЯ ВРЕМЕНИ ПОСЛЕДНЕЙ МОДИФИКАЦИИ И ИЗМЕНЕНИЯ INODE. ВРЕМЯ ПОСЛЕДНЕГО ДОСТУПА ПРИ ЭТОМ ОСТАЕТСЯ НЕИЗМЕННЫМ.

[9] В INODE УДАЛЯЕМОГО ФАЙЛА ОБНОВЛЯЕТСЯ ПОЛЕ DI_SPARE. В ИСХОДНЫХ ТЕКСТАХ ОНО ПОМЕЧЕНО КАК «RESERVED; CURRENTLY UNUSED», НО ПРОСМОТР ДАМПА ПОКАЗЫВАЕТ, ЧТО ЭТО НЕ ТАК. СУДЯ ПО ВСЕМУ, ЗДЕСЬ ХРАНИТСЯ НЕЧТО ВРОДЕ ПОСЛЕДОВАТЕЛЬНОСТИ ОБНОВЛЕНИЯ (UPDATE SEQUENCE), ИСПОЛЬЗУЕМОЙ ДЛЯ КОНТРОЛЯ ЦЕЛОСТНОСТИ INODE. НО ЭТО ТОЛЬКО ПРЕДПОЛОЖЕНИЕ.

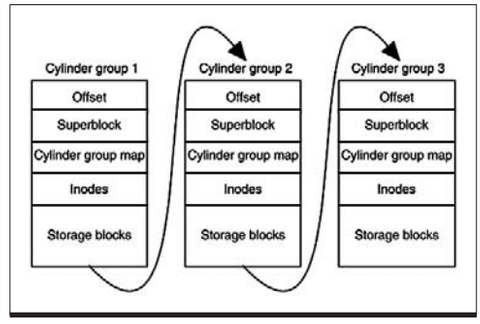
[10] В ДИРЕКТОРИИ УДАЛЕННОГО ФАЙЛА РАЗМЕР ПРЕДШЕСТВУЮЩЕЙ СТРУКТУРЫ DIRECT УВЕЛИЧИВАЕТСЯ НА D_RECLEN. В РЕЗУЛЬТАТЕ ЧЕГО ОНА КАК БЫ «ПОГЛОЩАЕТ» ИМЯ УДАЛЯЕМОГО ФАЙЛА, НО ЕГО ЗАТИРАНИЯ НЕ ПРОИСХОДИТ. ВО ВСЯКОМ СЛУЧАЕ, ОНО ЗАТИРАЕТСЯ НЕ СРАЗУ, А ТОЛЬКО ТОГДА, КОГДА В ЭТОМ ВОЗНИКНЕТ РЕАЛЬНАЯ НЕОБХОДИМОСТЬ.

→ **подготовка к восстановлению.** Если ты только что удалил файл, то лучшим способом восстановления будет RESET. Без шуток! Система сбрасывает дисковые буферы не сразу, а спустя некоторое время, поэтому своевременная перезагрузка или отключение питания спасут ситуацию, и после загрузки файл окажется целым и невредимым, правда, на самом диске могут образоваться значительные разрушения, так что риск неблагоприятного исхода очень велик, и лучше воспользоваться более традиционными средствами восстановления.

Первым делом размонтируй (unmount) дисковый раздел или перемонтируй его «только на чтение». Лечение активных разделов обычно заканчивается очень печально. Если восстанавливаемые файлы находятся в системном разделе, то можно прибегнуть к LiveCD. Лучше всего использовать KNOPPIX. Он поддерживает большое количество оборудования, не требователен к ресурсам (достаточно всего 128 Мб памяти) и содержит все необходимые утилиты для восстановления. Опытные пользователи могут сформировать загрузочный CD или даже дискету самостоятельно.

Широко разрекламированный дистрибутив Ferenzy 0.3, основанный на Free BSD, лучше сразу выкинуть в помойку — совсем немного дисковых утилит, да и те ориентированны в основном на ext2fs, а USF/FFS поддерживает постольку поскольку. Тем не менее, для восстановительных работ данный диск вполне пригоден, если ничего другого под рукой нет...

Дисковых редакторов, работающих на уровне секторов, под BSD существует не так уж и много. Обычно для этой цели пользуются BSD-портом LINUX-редактора lde (**http://lde.sourceforge.net**). Но, к сожалению, когда мы тестировали его на системе 4.5 BSD, он работал крайне нестабильно и не отображал основные структуры данных в удобочитаемом виде, хотя поддержка UFS в нем заявлена. В принципе, можно вставить в привод загрузочный CD-ROM с Windows PE и воспользоваться любым Windows-редактором, от Microsoft Disk Probe до Runtime Disk Explorer'a. То же самое справедливо и для Norton Disk Editor'a, запущенного с дискеты из-под MS-DOS (правда, ни диски большого объема, ни SCSI-устройства он не поддерживает). Еще можно запустить KNOPPIX или любой Live LINUX, ориентированный на восстановление, но дело в том, что, редактируя диск напрямую, его легко испортить. Одно неверное движение руки — и гигабайты данных обращаются в прах.



Последовательно расположенные группы цилиндров

При наличии свободного места рекомендуется создать копию раздела, и все дальнейшие опыты проводить уже над ней. В мире Windows для этой цели требуются специальные утилиты (например Norton Ghost), которые, кстати говоря, стоят нехилых денег. Но BDS — совсем другое дело. Здесь все необходимое находится под рукой. Копию раздела проще всего создать командой cp /dev/ad0s1a dump, где ad0s1a — имя устройства, а dump — имя файла-дампа, для работы с которым сгодится любой hex-редактор (например, biew — **http://biew.sourceforge.net**).

→ **структура UFS.** В начале диска расположен boot-сектор (на незагрузочных разделах он может быть пустым), а все остальное пространство поделено на несколько зон одинакового размера, называемых группами цилиндров (cylinder groups). Каждая группа цилиндров имеет свой суперблок (super-block), свою таблицу inode (Index-node)

НЕМНОГО ИСТОРИИ

UFS расшифровывается как UNIX File System и ведет свою историю от S5 FS — самой первой файловой системы, написанной для UNIX в далеком 1974 году. S5 FS была крайне простой и неповоротливой (по некоторым данным — 2%-5% от «сырой» производительности голого диска), но понятия суперблока (super-block), файловых записей (inodes) и блоков данных (blocks) в ней уже существовали.

В процессе работы над дистрибутивом 4.2 BSD, вышедшим в 1983 году, ординальная файловая система изменилась в лучшую сторону. Были добавлены длинные имена, символические ссылки и т. д. Так родилась UFS.

В 4.3 BSD, увидевшей свет уже в следующем году, улучшения носили более радикальный, если не сказать революционный, характер. Появились концепции фрагментов (fragments) и групп цилиндров (cylinder groups). Быстродействие файловой системы существенно возросло, что и определило ее название FFS — Fast File System (быстрая файловая система).

Все последующие версии линейки 4.x BSD прошли под знаменем FFS, но в 5.x BSD файловая система вновь изменилась. Для поддержки дисков большого объема ширину всех адресных полей пришлось удвоить: 32-битная нумерация фрагментов уступила место 64-битной. Были внесены и другие менее существенные усовершенствования.

Фактически мы имеем дело с тремя различными файловыми системами, не совместимыми друг с другом на уровне базовых структур данных. Однако некоторые источники склонны рассматривать FFS как надстройку над UFS. Из Little UFS2 FAQ следует, что «UFS/UFS2 определяет раскладку данных на диске. FFS реализована поверх UFS 1 или 2 и отвечает за структуру директорий и некоторых дисковых оптимизаций». Действительно, если заглянуть в исходные тексты файловой системы, то можно обнаружить два подкаталога — /ufs и /ffs. В /ffs находится определение суперблока (базовой структуры, отвечающей за раскладку данных), а в /ufs — определение inode и структуры директорий, что опровергает данный тезис, с точки зрения которого все должно быть с точностью до наоборот.

Чтобы не увязнуть в болоте терминологических тонкостей, под UFS мы будем понимать основную файловую систему 4.5 BSD, а под UFS2 — основную файловую систему 5.x BSD.

и свою группу блоков данных, совершенно независимую от всех остальных зон. Другим словами, inode описывает блоки данных той и только той зоны, к которой она принадлежит. Это увеличивает быстродействие файловой системы (головка жесткого диска совершает более короткие перемещения) и упрощает процедуру восстановления при значительном разрушении данных, поскольку, как показывает практика, обычно гибнет только первая группа inode. Чтобы погибли все группы... что же такого с жестким диском нужно сделать?! Под пресс положить, наверно.

Каждый блок, в свою очередь, разбит на несколько фрагментов фиксированного размера, предотвращающих потерю свободного пространства в хвостах файлов. Благодаря этому использование блоков большого размера уже не кажется расточительной идей: напротив, это увеличивает производительность и уменьшает фрагментацию. Если файл использует более одного фрагмента в двух несмежных блоках, он автоматически перемещается на новое место, в наименее фрагментированный регион свободного пространства. Поэтому фрагментация в UFS очень мала или же совсем отсутствует, что существенно облегчает восстановление удаленных файлов и разрушенных данных.

Адресация ведется либо по физическим смещениям, измеряемым в байтах и отсчитываемым от начала группы цилиндров (реже — от UFS-раздела), либо по номерам фрагментов, отсчитываемым от тех же самых точек. Допустим, размер блока составляет 16 Кб, разбитых на 8 фрагментов. Тогда сектор 69 будет иметь смещение 512 x 69 = 35328 байт или 1024 x (16/8)/512 x 69 = 276 фрагментов.

В USF первый суперблок располагается по смещению 8192 байт от конца загрузочного сектора, что соответствует 17-му сектору. В UFS2 он «переехал» на 65536 байт (129 секторов) от начала, освобождая место для дисковой метки и первичного загрузчика операционной системы. А для действительно больших (в исходных текстах — piggy, то есть «свинских») систем предусмотрена возможность перемещения суперблока по адресу 262144 байт (целых 513 секторов).

Вслед за суперблоком идут одна или несколько групп цилиндров, описываемых дескрипторами групп (group descriptors), карты свободного пространства (в просторечии — битмапы, block bitmap/inode bitmap) и таблицы inode. Для перестраховки копия суперблока дублируется в каждой группе. Загрузочный сектор не дублируется, но, по соображениям унификации, под него просто выделяется место. Таким образом, относительная адресация блоков в каждой группе остается неизменной.

Среди прочей информации суперблок содержит:
cblkno — смещение первой группы блока цилиндров, измеряемый в фрагментах, отсчитываемых от начала раздела;
fs_iblkno — смещение первой inode в первой группе цилиндров (фрагменты от начала раздела);
fs_dblkno — смещение первого блока данных в первой группе цилиндров (фрагменты от начала раздела);
fs_ncg — количество групп цилиндров (штуки);
fs_bsize — размер одного блока в байтах;
fs_fsize — размер одного фрагмента в байтах;
fs_frag — количество фрагментов в блоке;
fs_fpg — размер каждой группы цилиндров, выраженный в блоках (может быть найден через fs_cgsize);

Для перевода смещений, выраженных в фрагментах, в номера секторов служит следующая формула: sec_n(fragment_offset) = fragment_offset * (fs_bsize / fs_frag / 512). Или ее более короткая разновидность: sec_n(fragment_offset) = fragment_offset * fs_fsize / 512.

формат суперблока (определен в файле /src/ufs/ffs/fs.h, второстепенные поля опущены)

```
struct fs {
/* 0x00 */      int32_t fs_firstfield;
/* historic file system linked list, */
/* 0x04 */      int32_t fs_unused_1;
/*      used for incore super blocks */
/* 0x08 */      ufs_daddr_t fs_sblkno;
/* addr of super-block in filesys */
/* 0x0C */      ufs_daddr_t fs_cblkno;
/* offset of cyl-block in filesys */
/* 0x10 */      ufs_daddr_t fs_iblkno;
/* offset of inode-blocks in filesys */
/* 0x14 */      ufs_daddr_t fs_dblkno;
/* offset of first data after cg */
/* 0x18 */      int32_t fs_cgoffset;
/* cylinder group offset in cylinder */
/* 0x1C */      int32_t fs_cgmask;
/* used to calc mod fs_ntrak */
/* 0x20 */      time_t  fs_time;
/* last time written */
/* 0x24 */      int32_t fs_size;
/* number of blocks in fs */
/* 0x28 */      int32_t fs_dsize;
/* number of data blocks in fs */
/* 0x2C */      int32_t fs_ncg;
/* number of cylinder groups */
/* 0x30 */      int32_t fs_bsize;
/* size of basic blocks in fs */
/* 0x34 */      int32_t fs_fsize;
/* size of frag blocks in fs */
/* 0x38 */      int32_t fs_frag;
/* number of frags in a block in fs */
/* these are configuration parameters */
/* 0x3C */      int32_t fs_minfree;
/* minimum percentage of free blocks */
/* 0x40 */      int32_t fs_rotdelay;
/* num of ms for optimal next block */
/* 0x44 */      int32_t fs_rps;
/* disk revolutions per second */
/* sizes determined by number of cylinder
groups and their sizes */
/* 0x98 */      ufs_daddr_t fs_csaddr;
/* blk addr of cyl grp summary area */
/* 0x9C */      int32_t fs_cssize;
/* size of cyl grp summary area */
/* 0xA0 */      int32_t fs_cgsize;
/* cylinder group size */

/* these fields can be computed from
the others */
/* 0xB4 */      int32_t fs_cpg;
/* cylinders per group */
/* 0xB8 */      int32_t fs_ipg;
/* inodes per group */
/* 0xBC */      int32_t fs_fpg;
/* blocks per group * fs_frag */

/* these fields are cleared at mount time */
/* 0xD0 */      int8_t  fs_fmod;
/* super block modified flag */
/* 0xD1 */      int8_t  fs_clean;
/* file system is clean flag */
/* 0xD2 */      int8_t  fs_ronly;
/* mounted read-only flag */
/* 0xD3 */      int8_t  fs_flags;
/* see FS_ flags below */
/* 0xD4 */      u_char  fs_fsmnt
[MAXMNTLEN];    /* name mounted on */
};
```

В некотором отдалении от конца суперблока находится первая группа цилиндров. В начале каждой группы расположена служебная структура cg — описатель группы цилиндров, содержащая магическую последовательность 55h 02h 09h, по которой все уцелевшие группы можно найти даже при полностью испорченном суперблоке (стартовые адреса всех последующих групп вычисляются путем умножения номера группы на ее размер, содержащийся в поле fs_cgsize).

Другие важные параметры:
cg_cgx — порядковой номер группы, отсчитываемый от ноля;
cg_old_niblk — количество inode в данной группе;
cg_ndblk — количество блоков данных в рассматриваемой группе;
csum — количество свободных inode и блоков данных в рассматриваемой группе;
cg_iusedoff — смещение карты занятых inode, отсчитываемое от начала данной группы и измеряемое в байтах;
cg_freeoff — смещение карты свободного пространства (байты от начала группы).

структура описателя группы цилиндров (определена в файле /src/ufs/ffs/fs.h)

```
#define CG_MAGIC        0x090255
#define MAXFRAG 8
struct cg {
/* 0x00 */ int32_t      cg_firstfield;
/* historic cyl groups linked list */
/* 0x04 */ int32_t      cg_magic;
```

```
/* magic number */
/* 0x08 */ int32_t      cg_old_time;
/* time last written */
/* 0x0C */ int32_t      cg_cgx;
/* we are the cgx'th cylinder group */
/* 0x10 */ int16_t      cg_old_ncyl;
/* number of cyl's this cg */
/* 0x12 */ int16_t      cg_old_niblk;
/* number of inode blocks this cg */
/* 0x14 */ int32_t      cg_ndblk;
/* number of data blocks this cg */
/* 0x18 */ struct       csum cg_cs;
/* cylinder summary information */
/* 0x28 */ int32_t      cg_rotor;
/* position of last used block */
/* 0x2C */ int32_t      cg_frotor;
/* position of last used frag */
/* 0x30 */ int32_t      cg_irotor;
/* position of last used inode */
/* 0x34 */ int32_t      cg_frsum[MAX-
FRAG]; /* counts of available frags */
/* 0x54 */ int32_t      cg_old_btotoff;
/* (int32) block totals per cylinder */
/* 0x58 */ int32_t      cg_old_boff;
/* (u_int16) free block positions */
/* 0x5C */ int32_t      cg_iusedoff;
/* (u_int8) used inode map */
/* 0x60 */ int32_t      cg_freeoff;
 /* (u_int8) free block map */
/* 0x64 */ int32_t      cg_nextfreeoff;
/* (u_int8) next available space */
/* 0x68 */ int32_t      cg_clustersu-
moff;/*  (u_int32) counts of avail clu-
sters */
/* 0x6C */ int32_t      cg_clusteroff;
/* (u_int8) free cluster map */
/* 0x70 */ int32_t      cg_nclusterblks;
/* number of clusters this cg */
/* 0x74 */ int32_t      cg_niblk;
/* number of inode blocks this cg */
/* 0x78 */ int32_t      cg_initediblk;
/* last initialized inode */
/* 0x7C */ int32_t      cg_spare-
con32[3];  /* reserved for future use */
/* 0x00 */ ufs_time_t   cg_time;
/* time last written */
/* 0x00 */ int64_t      cg_spare-
con64[3];   /* reserved for future use */
/* 0x00 */ u_int8_t     cg_space[1];
/* space for cylinder group maps */
/* actually longer */
```

Между описателем группы цилиндров и группой inode расположена карта занятых inode и карта свободного дискового пространства, которые представляют собой обыкновенные битовые поля, точно такие же, как и в NTFS. При восстановлении удаленных файлов без этих карт никуда! Они существенно сужают круг поиска, что особенно хорошо заметно на дисках, заполненных более чем наполовину.

За картами следует массив inode, смещение которого содержится в поле cg_iusedoff (адрес первой группы inode продублирован в суперблоке). В UFS inode играет ту же самую роль, что и FILE Record в NTFS (в FAT прямых аналогов нет). Здесь сосредоточена вся информация о файле: тип файла (обычный файл, директория, символьная ссылка и т.д.), логический и физический размер, схема размещения на диске, время создания, модификации, последнего доступа и удаления, права доступа и количество ссылок на файл.

По сути, в UFS структура inode ничем не отличается от ext2fs, только расположение полей другое. К тому же имеется только один блок косвенной адресации вместо трех, но это уже детали, в которые не будем углубляться (иначе или зависнем, или завязнем). Лучше рассмотрим назначение фундаментальных полей, к числу которых принадлежат:

di_nlink — количество ссылок на файл (0 означает «удален»);

di_size — размер файла в байтах;

di_atime/di_atimensec — время последнего доступа к файлу;

di_mtime/di_mtimensec — время последней модификации;

di_ctime/di_ctimensec — время последнего изменения inode;

di_db — адреса первых 12-ти блоков данных файла, отсчитываемые в фрагментах от начала группы цилиндров;

di_ib — адреса блоков косвенной адресации (фрагменты от начала группы).

структура inode в USF1 (определена в файле /src/ufs/ufs/dinode.h)

```
struct dinode {
/* 0x00 */   u_int16_t     di_mode;
/*   0: IFMT, permissions; see below. */
/* 0x02 */   int16_t       di_nlink;
/*   2: File link count. */
/* 0x04 */   union {u_int16_t oldids[2];
/*   4: Ffs: old user and group ids.
*/           int32_t       inumber;
/*   4: Lfs: inode number. */
             } di_u;
/* 0x08 */   u_int64_t     di_size;
/*   8: File byte count. */
/* 0x10 */   int32_t       di_atime;
/*  16: Last access time. */
/* 0x14 */   int32_t       di_atimensec;
/*  20: Last access time. */
/* 0x18 */   int32_t       di_mtime;
/*  24: Last modified time. */
/* 0x1C */   int32_t       di_mtimensec;
/*  28: Last modified time. */
/* 0x20 */   int32_t       di_ctime;
/*  32: Last inode change time. */
/* 0x24 */      int32_t di_ctimensec;
/*  36: Last inode change time. */
/* 0x28 */      ufs_daddr_t di_db[NDADDR];
/*  40: Direct disk blocks. */
/* 0x58 */      ufs_daddr_t di_ib[NIADDR];
/*  88: Indirect disk blocks. */
/* 0x64 */      u_int32_t   di_flags;
/* 100: Status flags (chflags). */
/* 0x68 */      int32_t     di_blocks;
/* 104: Blocks actually held. */
/* 0x6C */      int32_t     di_gen;
/* 108: Generation number. */
/* 0x70 */      u_int32_t   di_uid;
/* 112: File owner. */
/* 0x74 */      u_int32_t   di_gid;
/* 116: File group. */
/* 0x78 */      int32_t     di_spare[2];
/* 120: Reserved; currently unused */
};
```

В UFS2 формат inode был существенно изменен — появилось множество новых полей, удвоилась ширина адресных полей и т. д. Что это обозначает в практическом плане? Смещения всех полей изменились, только и всего, а общий принцип работы остался прежним.

структура inode в USF2

```
struct ufs2_dinode {
/* 0x00 */ u_int16_t        di_mode;
/*   0: IFMT, permissions; see below. */
/* 0x02 */ int16_t          di_nlink;
/*   2: File link count. */
/* 0x04 */ u_int32_t        di_uid;
/*   4: File owner. */
/* 0x08 */ u_int32_t        di_gid;
/*   8: File group. */
/* 0x0C */ u_int32_t        di_blksize;
/*  12: Inode blocksize. */
/* 0x10 */ u_int64_t        di_size;
/*  16: File byte count. */
/* 0x18 */ u_int64_t        di_blocks;
/*  24: Bytes actually held. */
/* 0x20 */ ufs_time_t       di_atime;
/*  32: Last access time. */
/* 0x28 */ ufs_time_t       di_mtime;
/*  40: Last modified time. */
/* 0x30 */ ufs_time_t       di_ctime;
/*  48: Last inode change time. */
/* 0x38 */ ufs_time_t       di_birthtime;
/*  56: Inode creation time. */
/* 0x40 */ int32_t          di_mtimensec;
/*  64: Last modified time. */
/* 0x44 */ int32_t          di_atimensec;
/*  68: Last access time. */
/* 0x48 */ int32_t       di_ctimensec;
/*  72: Last inode change time. */
/* 0x4C */ int32_t       di_birthnsec;
/*  76: Inode creation time. */
/* 0x50 */ int32_t       di_gen;
/*  80: Generation number. */
/* 0x54 */ u_int32_t     di_kernflags;
/*  84: Kernel flags. */
/* 0x58 */ u_int32_t     di_flags;
```

```
/*  88: Status flags (chflags). */
/* 0x5C */ int32_t      di_extsize;
/*  92: External attributes block. */
/* 0x60 */ ufs2_daddr_tdi_extb[NXA-
DR];/*  96: External attributes block. */
/* 0x70 */ ufs2_daddr_tdi_db[NDADDR];
/* 112: Direct disk blocks. */
/* 0xD0 */ ufs2_daddr_tdi_ib[NIADDR];
/* 208: Indirect disk blocks. */
/* 0xE8 */ int64_t      di_spare[3];
/* 232: Reserved; currently unused */
 };
```

Имена файлов хранятся в директориях. В inode их нет. С точки зрения UFS, директории являются обыкновенными файлами (ну, не совсем обыкновенными) и могут храниться в любом месте, принадлежащем группе цилиндров. Файловая система UFS поддерживает несколько типов хэширования директорий, однако на структуре хранения имен это никак не отражается. Имена хранятся в блоках, называемых DIRBLKSIZ в структурах типа direct, выровненных по 4-байтовой границе.

Структура direct определена в файле /src/ufs-/ufs/dir.h и содержит: номер inode, описывающий данный файл, тип файла, его имя, а также длину самой структуры direct, используемую для нахождения следующего direct'a в блоке.

структура direct, отвечающая за хранение имен файлов и директорий

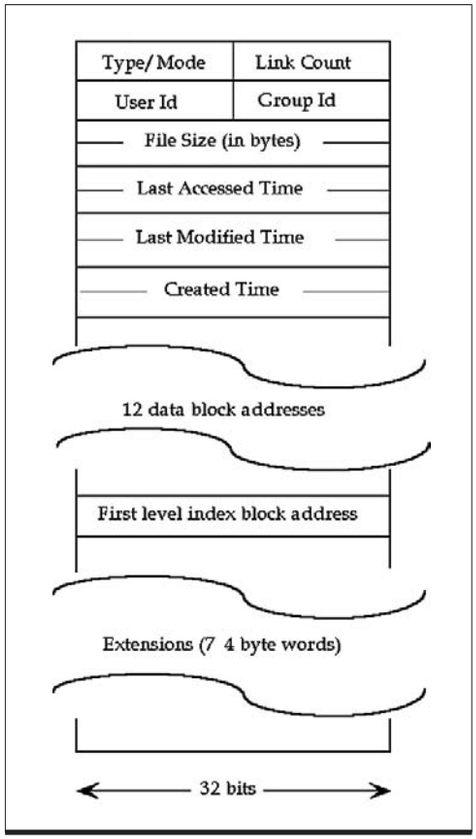
```
struct  direct {
/* 0x00 */ u_int32_t    d_ino;
  /* inode number of entry */
/* 0x04 */ u_int16_t    d_reclen;
/* length of this record */
/* 0x06 */ u_int8_t     d_type;
/* file type, see below */
/* 0x07 */ u_int8_t     d_namlen;
/* length of string in d_name */
/* 0x08 */ char d_name[MAXNAMLEN + 1];
/* name with length <= MAXNAMLEN */
};
```

➔ **техника ручного восстановления файлов.** Начнем с грустного. Поскольку при удалении файла ссылки на 12 первых блоков и 3 блока косвенной адресации необратимо затираются, автоматическое восстановление данных невозможно в принципе. Найти удаленный файл можно только по его содержимому. Искать, естественно, необходимо в свободном пространстве. Вот тут-то нам и пригодятся карты, расположенные за концом описателя группы цилиндров.

Если нам повезет, и файл окажется нефрагментированным (а на UFS, как уже отмечалось, фрагментация обычно отсутствует или крайне невелика), то остальное будет делом техники. Просто выделяешь группу секторов и записываешь ее на диск, но только ни в коем случае не на сам восстанавливаемый раздел! К примеру, файл можно передать на соседнюю машину по сети. К сожалению, поле длины файла безжалостно затирается при его удалении, и актуальный размер приходится определять «на глазок». Звучит страшнее, чем выглядит. Неиспользуемый хвост последнего фрагмента всегда забивается нулями, что дает хороший ориентир. Проблема в том, что некоторые типы файлов содержат в своем конце некоторое количество нулей, при отсечении которых их работоспособность нарушается, поэтому тут приходится экспериментировать.

А если файл фрагментирован? Первые 13 блоков (именно блоков, а не фрагментов!) придется собирать руками. В идеале это будет один непрерывный регион. Хуже, если первый фрагмент расположен в «чужом» блоке (то есть блоке, частично занятом другим файлом), а оставшиеся 12 блоков находятся в одном или нескольких регионах. Вообще-то, достаточно трудно представить себе ситуацию, в которой первые 13 блоков были бы сильно фрагментированы (а поддержка фоновой дефрагментации в UFS на что?!). Такое может произойти только при интересной «перегруппировке» большого количеств файлов, что в реальной жизни практически никогда не встречается, разве только если ты задумал навести порядок на своем жестком диске. Короче, будем считать, что 13-ый блок файла найден. В массив непосредственной адресации он уже не влезает (там содержатся только 12 блоков), и ссылка на него, как и на все последующие блоки файла, должна содержаться в блоках косвенной адресации, которые при удалении файла помечаются как свободные, но не затираются. Точнее затираются, но не сразу. Большинство файлов обходятся только одним косвенным блоком, что существенно упрощает нашу задачу.

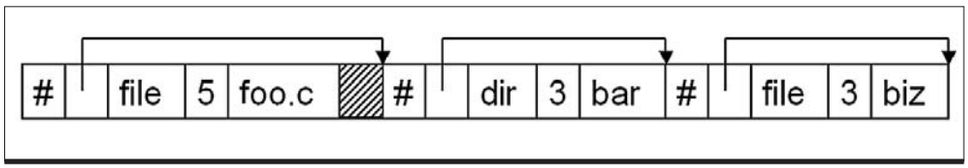
Как найти этот блок на диске? Вычисляем смещение 13-го блока файла от начала группы цилиндров, переводим его в фрагменты, записываем получившееся число задом наперед (так, чтобы младшие байты располагались по меньшим адресам) и осуществляем контекстный поиск в свободном пространстве. Отличить блок косвенной адресации от всех остальных типов данных очень легко — он представляет собой массив указателей на блоки, а в конце идут нули. Остается только извлечь эти блоки с диска и записать их в файл, обрезая его по нужной длине. Внимание! Если ты нашел несколько «кандидатов» в блоки косвенной адресации, это означает, что 13-ый блок удаленного файла в разное время принадлежал различным файлам (а так, скорее всего, и будет). Не все косвенные блоки были затерты — вот ссылки и остались.



Схематичное изображение inode

Как отличить «наш» блок от «чужих»? Если хотя бы одна из ссылок указывает на уже занятый блок данных (что легко определить по карте), то такой блок можно сразу откинуть. Оставшиеся блоки перебираются вручную до получения работоспособной копии файла. Имя файла, если оно еще не затерто, можно извлечь из директории. Естественно, при восстановлении нескольких файлов ты не можешь однозначно сказать, какое из имен какому файлу принадлежит. Тем не менее, это все же лучше, чем совсем ничего. Директории восстанавливаются точно так же, как и обыкновенные файлы, хотя, по правде говоря, в них кроме имен файлов нечего восстанавливать...

➔ **заключение.** В описанном методе восстановления данных — много ограничений. В частности, при удалении большого количества сильно фрагментированных двоичных файлов, он говорит «пас» и уходит в кусты. Ты только убьешь свое время, но вряд ли найдешь среди обломков файловой системы что-то полезное. Но как бы там ни было, другого выхода просто нет (если, конечно, не считать резервной копии, которой тоже нет) **C**



Хранение имен файлов и директорий

SPYWARE

В СЛЕДУЮЩЕМ НОМЕРЕ МЫ РАСКРОЕМ ТАЙНЫ BSD-СИСТЕМ:
МОБИЛЬНОГО СПАМА, SPYWARE, ТРОЯНОВ, БОТНЕТОВ, ФИШИНГА, А ТАКЖЕ РАССКАЖЕМ, КАК ОТ ВСЕГО ЭТОГО ЗАЩИТИТЬСЯ!

СКОРО В СПЕЦЕ:
WINDOWS VISTA
ВЗГЛЯД ИЗНУТРИ. ПОДРОБНЫЙ АНАЛИЗ НОВОЙ ОС ОТ MICROSOFT. НОВЕЙШИЕ ТЕХНОЛОГИИ. УДОБСТВО, БЫСТРОТА РАБОТЫ.
ИСКУССТВО ПРОГРАММИРОВАНИЯ
АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ. СОВЕТЫ БЫВАЛЫХ КОДЕРОВ. СРЕДСТВА РАЗРАБОТКИ. ОТЛАДЧИКИ И ДИЗАССЕМБЛЕРЫ.

SPECIAL ИНТЕРВЬЮ

Как мы впервые увидились с Андрюшком? Давно это было. В те легендарные времена, когда люди были, как братья, и деревья могли разговаривать, сидели мы в старом офисе, в тени раскидистого баобаба и попивали чай с вареньем. На улице была сильная гроза, и мы не сразу поняли природу внезапно раздавшегося звука. Помню, Покровский тогда сразу извлек из-под стола осиновый кол и серебряные пули (мы тогда увлекались «Молотом Ведьм» на ночь), распознав в этом звуке агрессивного обладателя железных копыт. Но, как оказалось, наш инквизиторский стафф был неэффективен против рогатого гостя — ведь он не настоящий демон, а символический. Веришь или нет, но он — самый настоящий BSD-механизм. Попробуем проникнуть в его биографию.

НАЧНЕМ С САМОГО ИНТЕРЕСНОГО ВОПРОСА И ПЛАВНО ПЕРЕЙДЕМ К ТВОЕЙ ЛИЧНОСТИ. ИТАК, КАК ТЫ ОТНОСИШЬСЯ К НАШЕМУ ЖУРНАЛУ? ЧИТАЕШЬ СПЕЦ, НРАВИТСЯ ЛИ ОН ТЕБЕ?

АНДРЕЙ МАТВЕЕВ: Честно говоря, я не имею возможности читать каждый номер Спеца. Но те номера, которые видел, произвели на меня приятное впечатление. Над двумя выпусками («UNIX без проблем» и «Неприступный UNIX») даже посчастливилось поработать. Несмотря на столь юный, по издательским меркам, возраст, журнал довольно успешно развивается. Вы многого добились, но заветная планка, поставленная «Хакером», вам еще не покорилась. Поэтому хочу пожелать развития и повышения тиража.

У ЧИТАТЕЛЕЙ ЕСТЬ ПОДОЗРЕНИЕ, ЧТО АНДРЮШОК — НЕ НАСТОЯЩИЙ ЧЕЛОВЕК, А СИНТЕТИЧЕСКИЙ. ЧТО ТЫ НА ЭТО СКАЖЕШЬ?

АНДРЕЙ МАТВЕЕВ: На этот счет существует даже целая теория. Будто бы меня (компьютерную систему, имитирующую человеческое мышление) создали в недрах сверхсекретной лаборатории, находящейся в ведомстве министерства энергетики США. Спешу разочаровать тебя и читателей — это не так, я обычный человек с присущими ему пороками и недостатками. Хотя иногда мне кажется, что я веду синтетический образ жизни, и мне снятся синтетические сны...

ЭТО ВСЕ ПОНЯТНО: ГОВОРЯТ, ВСЕ ИНОПЛАНЕТЯНЕ, КИБОРГИ И ПРОЧИЕ ГУМАНОИДЫ ИМЕННО ТАК О СЕБЕ И ПОВЕСТВУЮТ. НО БУДЕМ СЧИТАТЬ, МЫ ТЕБЕ ПОВЕРИЛИ. НУ ЧТО, КОРОТКО О СЕБЕ? КТО ТАКОЙ, ЧЕМ ЗАНИМАЕШЬСЯ? У ТЕБЯ, НАВЕРНОЕ, СЕМЬЯ, ДЕТИ, ДРУЗЬЯ ЕСТЬ? РАССКАЖИ-КА О СВОЕМ ЖИТЬЕ-БЫТЬЕ.

АНДРЕЙ МАТВЕЕВ: Родился в 1979 году в Нижнем Новгороде. Прожив на родине Максима Горького шесть лет, с родителями переехал в столицу, где закончил сначала школу, а затем университет и аспирантуру. Друзья-приятели совершенно разные, друг на друга не похожие, но с одинаковыми тараканами. Не женат, но я работаю в этом направлении.

НУ, НЕ БУДЕМ РАССУЖДАТЬ О ТОМ, ЧЕМ И КАК ТЫ ТРУДИШЬСЯ В ЭТОМ НАПРАВЛЕНИИ, ЛУЧШЕ РАССКАЖИ ПРО НАСТОЯЩУЮ РАБОТУ. НАСЧЕТ ТОГО, ЧТО ТЫ РЕДАКТОР РУБРИКИ UNIXOID ХАКЕР'А, МЫ В КУРСЕ, НО ВОТ Я СЛЫШАЛ, ЧТО ТЫ РАБОТАЕШЬ В КАКОМ-ТО МОГУЧЕМ НИИ? ЭТО ПРАВДА?

АНДРЕЙ МАТВЕЕВ: Моя основная работа — заведующий IT-отделом во Всероссийском НИИ, сотрудники которого занимаются научным обоснованием развития сельскохозяйственной мелиорации и гидротехники. Вообще, этот институт просто потрясающий, настоящий храм науки. Никитос и Токса, когда заходили в гости, несказанно впечатлились увиденным. Я не открою большой тайны, если скажу, что сейчас зарплата в бюджетных конторах только начинает приближаться к чисто символической, поэтому по совместительству работаю системным администратором в нескольких коммерческих фирмах. Кроме того, как ты уже упомянул, work'аю над журналом «Хакер». А на досуге для знакомых поднимаю виртуальные частные сети по протоколу IPSec и ваяю/поддерживаю сайты на базе CMS.

ОГО, КАКОЕ ЖЕ У ТЕБЯ ОБРАЗОВАНИЕ?

АНДРЕЙ МАТВЕЕВ: За шесть лет обучения в университете я получил два диплома о высшем образовании (дипломы бакалавра и магистра по специальности «Строительное дело»). Мне несколько раз предлагали работать по специальности, но я смутно представляю себя прорабом или

Андрей Матвеев — заведующий IT-отделом во Всероссийском НИИ. Редактор рубрики «юниксоид» журнала «Хакер».

сотрудником проектной организации, да и теперь уже совершенно точно уверен в том, что это не мое. Кстати, в моих статьях для «Хакера» периодически проскальзывают фразы типа: «закладываем надежный фундамент для нашей конструкции».

ВООБЩЕ, СПОРНЫЙ ВОПРОС. ПО-МОЕМУ, РАБОТАТЬ СТРОИТЕЛЕМ — КРУТО. КАК У ГРУППЫ «КУВАЛДА» — «БЕТОНОМЕШАЛКА МЕШАЕТ БЕТОН, БРИГАДА СТРОИТЕЛЕЙ ЖРЕТ САМОГОН». ОПЯТЬ ЖЕ, ВИЛЛЫ БЫ СТРОИЛ, ГАСТРБАЙТЕРОВ ГОНЯЛ. ХОТЯ БЫТЬ ХАРДКОРНЫМ ЮНИКСОИДОМ ТОЖЕ ОЧЕНЬ ЗДОРОВО. РАССКАЖИ, ЧЕМ ТЫ ВООБЩЕ КРУТ В ЭТОМ ПЛАНЕ? ПОЧЕМУ МЫ РЕШИЛИ БРАТЬ ИНТЕРВЬЮ У ТЕБЯ, А НЕ У БИЛЛА ГЕЙТСА? ХОДИТ СЛУХ, ЧТО ТЫ КРУТОЙ КЕРНЕЛ-ХАКЕР И ЧУТЬ ЛИ НЕ ОДИН ИЗ РАЗРАБОТЧИКОВ OPENBSD.

АНДРЕЙ МАТВЕЕВ: Постараюсь немного прояснить ситуацию. Начиная с конца 2002 года, в OpenBSD было внесено порядка двухсот моих изменений, и всего 20 из них касались ядра операционной системы, так что меня с огромным трудом можно назвать кернел-хакером. Я уделял первостепенное внимание аудиту исходного кода и по разным причинам старался не привносить в kernel/userland новый функционал. В большинстве своем мои патчи представляли собой наборы исправлений, которые были направлены на устранение ошибок (fd/FILE */memory leaks, double free, integer overflow, format string bug и др.) в библиотеке libc, сетевых демонах и штатных утилитах. Вопреки слухам, я не являюсь официальным разработчиком OpenBSD, хотя речь об этом неоднократно заходила с лидером проекта (Theo de Raadt).

ТАК ЗНАЧИТ, ПРИХОДИЛОСЬ ОБЩАТЬСЯ С РАЗРАБОТЧИКАМИ? РАССКАЖИ О САМЫХ КОЛОРИТНЫХ.

АНДРЕЙ МАТВЕЕВ: По электронной почте и на канале #hackers приватного чат-сервера мне удалось пообщаться со многими ведущими разработчиками OpenBSD (чат представляет собой нечто вроде урезанного по возможностям irc, потусить на нем можно только по спецприглашению). Довольно проблематично выделить из них кого-то одного, могу лишь отметить, что искрометный юмор на технические темы присутствует в изобилии. Цитата для примера: «You are in a maze of gpio pins, all alike, all undocumented, and a few are wired to bombs». В реале несколько раз встречался с Александром Юрченко (grange@), который занимается преимущественно дисковой подсистемой.

ИТОГО... 180 ИЗМЕНЕНИЙ В СИСТЕМНЫХ БИБЛИОТЕКАХ И ПРОГРАММАХ ИЗ ПРОСТРАНСТВА ПОЛЬЗОВАТЕЛЯ, ПЛЮС 20 ИЗМЕНЕНИЙ В ЯДРЕ... ХИЛО ПОЛУЧАЕТСЯ. ДАЖЕ МОЯ БАБУШКА ВНЕСЛА БОЛЬШЕ. РАССКАЖИ ПРО ПОСЛЕДНИЙ ПОДВИГ НА НИВЕ BSD.

АНДРЕЙ МАТВЕЕВ: В конце прошлого года я прикупил себе очередную игрушку для души. На этот раз ей стала ультракомпактная, нетребовательная к питанию и абсолютно бесшумная система, представляющая собой материнскую плату VIA Epia MS форм-фактора mini-ITX с безвентиляторным процессором VIA Eden ESP. Висевшая на борту 100 мегабитная сетевуха с внешним PHY VT6103, наотрез отказалась работать. Три строчки кода, добавленные в ядерный драйвер vr(4), оказали на нее целительное воздействие. Brad Smith, один из девелоперов, заметил мой интерес к карточкам на чипсетах VIA Rhine, и мы с ним примерно за два месяца привели этот драйвер в порядок — вычистили неиспользуемый код, реорганизовали проверки, вернули потерянные при портировании из FreeBSD куски кода на свои места, сделали более сглаженными вход и выход из неразборчивого режима (promiscuous mode) и т.д. В настоящее время отлавливаю утечки памяти в демоне isakmpd(8), который обеспечивает работу по протоколу обмена секретными ключами.

КСТАТИ, А ЕСТЬ ИНФОРМАЦИЯ О ТОМ, КТО СЕЙЧАС ИСПОЛЬЗУЕТ OPENBSD?

АНДРЕЙ МАТВЕЕВ: Благодаря своей многоплатформенности, надежности, масштабируемости и секьюрности за последние несколько лет OpenBSD снискала себе огромную популярность и нашла широкое применение как у простых пользователей, так и у крупных провайдеров. Более того, самое пристальное внимание на эту ось стали обращать правительственные организации. Публично о переводе части своих серверов под управление OpenBSD заявили правительства Австралии, Мексики и Чили. А если говорить о пакете сетевых инструментов OpenSSH, любимом детище разработчиков OpenBSD, то его используют абсолютно все, кому нужен защищенный доступ к удаленным компьютерам.

УЧАСТИЕ В РАЗРАБОТКЕ САМОЙ ЗАЩИЩЕННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ В МИРЕ — ЭТО, КОНЕЧНО, ХОРОШО, А КАК ОБСТОЯТ ДЕЛА С OPENBSD.RU? ЭТО ТВОЙ ПРОЕКТ?

АНДРЕЙ МАТВЕЕВ: Основателем OpenBSD.ru и идейным вдохновителем всех русскоязычных пользователей OpenBSD является Олег Сафиуллин (form@). Я всего лишь один из 12 участников этого проекта. Что касается сайта **www.openbsd.ru**, то сейчас активно ресурс поддерживают 3 человека, еще несколько постоянных авторов пишут материалы и переводят официальную документацию. За последний год было подготовлено около 40 пошаговых руководств для всех уровней подготовки: от новичка до эксперта. Но проект всегда славился не столько своими переводами и статьями, сколько списком рассылки openbsd@openbsd.ru, где оперативно можно получить технически грамотную консультацию и практические рекомендации по любому вопросу, касающемуся ОС OpenBSD.

ОК, С OPENBSD МЫ РАЗОБРАЛИСЬ. ИДЕМ ДАЛЬШЕ. НЕ СЕКРЕТ, ЧТО ЖУРНАЛЫ «СПЕЦХАКЕР» И «ЖЕЛЕЗО» СТАЛИ ДОВОЛЬНО УСПЕШНЫМИ ПРОЕКТАМИ. КАК НАСЧЕТ ТОГО, ЧТОБЫ РАСШИРИТЬ СЕРИЮ И СДЕЛАТЬ ЖУРНАЛ «ЮНИКСОИД»?

АНДРЕЙ МАТВЕЕВ: Попробую восстановить цепь событий, произошедших примерно полтора года назад. На одной из редколлегий выяснилось, что в плане издательства «Gameland» присутствует графа «создание журнала, посвященного *nix-тематике». Я взялся за разработку концепции такого периодического издания, подобрал материалы, получил согласие на участие в проекте от своих постоянных авторов (это 10 человек, которые пишут в Xakep/Unixoid). Когда у меня в голове сформировалось четкое представление о том, каким должен быть новый журнал, я написал одному из издателей Геймленда. В своем ответе Сергей Покровский сослался на отсутствие рекламодателей и низкую заинтересованность у целевой аудитории в покупке печатной версии журнала. Но вместе с отказом осторожно обнадежил, сказав, что к этому разговору можно будет вернуться в конце 2006 года.

Также витала в облаках идея создания журнала без патронажа какого-либо московского издательства, другими словами, на собственные средства и вливания меценатов/заинтересованных лиц (здесь нельзя не вспомнить одну девушку, которая собиралась пожертвовать $1000 своих кровных). Но как мы не пересчитывали смету, как не урезали все до минимума, ежемесячные убыт-

ки должны были составить от 8 до 12 тысяч вечнозеленых. Да, нам было под силу выпустить два-три пилотных номера. Но довольно сложно создавать креатив, зная, что проект ждет неминуемый финансовый крах.

Так что пока журналу «Юниксоид» не суждено появиться на свет.

В НАЧАЛЕ НАШЕЙ БЕСЕДЫ ТЫ СКАЗАЛ, ЧТО ЗАКОНЧИЛ АСПИРАНТУРУ. НАВЕРНОЕ, И КАНДИДАТСКУЮ ДИССЕРТАЦИЮ УЖЕ НАПИСАЛ?

АНДРЕЙ МАТВЕЕВ: Ох, кандидатскую диссертацию пытаюсь одолеть на протяжении пяти лет. Работа, различные дела и всякие мелочи невероятно отвлекают от написания. Каркас диссера и математическая модель готовы, научная новизна и практическая значимость присутствуют, результаты были доложены на секциях ученого совета и научно-практических конференциях. Если совсем вкратце, то цель исследования состоит в разработке элементов информационной технологии управления агромелиоративными режимами на локальном (применительно к системам точного земледелия) и региональном (сетевое информационное обеспечение сельскохозяйственного производства) уровнях.

МОЩНО! А ПИСАЛ ЛИ ТЫ КОГДА-НИБУДЬ СТАТЬИ ПО НАУЧНОЙ ТЕМАТИКЕ?

АНДРЕЙ МАТВЕЕВ: Да, основные положения диссертационной работы были опубликованы в восьми печатных статьях. Но не думаю, что читателям СпецХакера будут интересны подробности из статьи «Аппроксимация влияния агроэкологических факторов на продуктивность агроценоза» или «Повышение эффективности управления информационными ресурсами в гидромелиорации», поэтому давай отойдем от этой темы.

ПРО ТВОЕ НАСТОЯЩЕЕ ВСЕ ЯСНО, А ЧТО ЖЕ С ТВОИМ ПРОШЛЫМ? КОГДА ТЫ ВООБЩЕ ПОЗНАКОМИЛСЯ С ЭВМ, КАК ВОЗЛЮБИЛ СЕЙ АППАРАТ И КАК ВПЕРВЫЕ ПОЗНАКОМИЛСЯ С UNIX? РАССКАЖИ-КА О СВОЕМ ПЕРВОМ СВИДАНИИ.

АНДРЕЙ МАТВЕЕВ: Насколько я помню, мое первое знакомство с компьютером состоялось в 1990 году. В зените перестройки моя маман работала оператором АСУ в бюро по туризму и экскурсиям (да-да, на летние каникулы меня отправляли в Сочи и дружественную Болгарию, а не в пионерский лагерь «Призрак кумача». Когда после школы я приходил к ней на работу, вся мощь компьютера с процессором 80286 без промедления направлялась на прохождение уровней «Принца Персии» и сбор алмазов в «Диггере». Я был до глубины души потрясен 16-цветовой картинкой и пищанием PC-спикера. В авральные дни, когда меня не подпускали к компу, я любил наблюдать за процессом бэкапа на магнитные ленты. С UNIX впервые столкнулся в 1998 году, когда принимал посильное участие в Российско-Германском проекте Ока-Эльба. Исследования загрязнения тяжелыми металлами донных отложений проводились на рабочей станции Sun Sparc с предустановленной Solaris 2.5.1.

И ЧТО, ПРОНИКСЯ К UNIX? ЧТО ЖЕ БЫЛО ДАЛЬШЕ, НА ДОМАШНЕМ КОМПЕ, НА РАБОТЕ?

АНДРЕЙ МАТВЕЕВ: На первый взгляд, все в системе показалось каким-то чуждым и даже диким. Интерес к изучению подогревался тем обстоятельством, что документации по UNIX и специалистов в данной области было в то время крайне мало. Здесь определенную роль сыграла моя детская мечта — изучить язык, понятный только посвященным. Плюс к этому, каждая успешно выполненная серия операций приводила в искренний восторг, и казалось, что возможности для личного и карьерного роста практически неисчерпаемы. В конце 2000 года меня подрядили на администрирование Linux-серверов. Затем были первые шаги в FreeBSD. Знакомство с фрей осталось шапочным, так как мой выбор пал на OpenBSD. Дома, листая книжки и просматривая исходный код, я разбирался с внутренним устройством *nix-систем и пытался понять механизмы функционирования. На работе экспериментировал с различными видами защиты, а поздними вечерами, когда сотрудники покидали институт, устраивал сетевые баталии: снифал трафик и проводил атаки типа denial-of-service, tcp-hijacking, ip-spoofing. Как ты понимаешь, исключительно в образовательных целях.

А ЧТО БЫ ТЫ ХОТЕЛ ИЗМЕНИТЬ В СВОЕМ КОМПЬЮТЕРНОМ ПРОШЛОМ, ЕСЛИ БЫ СЕЙЧАС ТЕБЕ ПРЕДСТАВИЛСЯ ТАКОЙ ШАНС?

АНДРЕЙ МАТВЕЕВ: Я постарался бы уделить самое пристальное внимание изучению английского языка, прикладной математики, программирования (на Asm/C/C++) и практической криптографии. В последней области, к сожалению, обладаю крайне поверхностными знаниями. А ведь в ней есть поистине неповторимое очарование. Чего только стоят блочные шифры, функции хэширования, коды аутентичности сообщений...

А ЧТО ТЫ МОЖЕШЬ ПОСОВЕТОВАТЬ НАЧИНАЮЩЕМУ ЮНИКСОИДУ? КАКИЕ ПОДВОДНЫЕ КАМНИ ЖДУТ НА ПУТИ? ОТ ЧЕГО МОЖНО ПРЕДОСТЕРЕЧЬ?

АНДРЕЙ МАТВЕЕВ: Наверное, начинающему изучать *nix можно посоветовать запастись терпением и любознательностью. Также не стоит из Windows XP сразу бросаться в FreeBSD: подобные переходы зачастую бывают неудачными. Как правило, тернистый путь юниксоида выглядит следующим образом: Windows → Linux (rpm based) → Slackware Linux → FreeBSD (→ OpenBSD). Что касается Linux, то ни один из существующих на сегодняшний день дистрибутивов не имеет преимущества над другими по всем показателям. Постарайся использовать дистрибутив, который посоветовал человек, способный потом помочь в его освоении. Если ты новичок, и знакомых гуру нет в радиусе нескольких километров, попробуй Mandriva или Fedora Core. Не стесняйся шерстить поисковики — документации сейчас предостаточно. И совсем необязательно, чтобы она была на русском или английском языке. При наличии в тексте команд и названий программ можно восстановить смысл любой статьи, посвященной *nix, и не важно, написана она японскими иероглифами или арабской вязью. Предложу еще две небольшие рекомендации. Во избежание потери данных экспериментируй с *nix на отдельном винчестере. И на первых порах устанавливай все предлагаемые пакеты. Потом, после настройки, ты всегда успеешь удалить лишнее.

«Как скомпилировать последнюю версию архинужной программы, как изменить менеджер окон, как эффективно работать в консоли?» — все эти и многие другие вопросы волнуют начинающих в мире *nix. Искать ответы на них лучше самостоятельно, читая документацию и закрепляя полученные знания на практике. Тогда это будет действительно результативно.

И, НАКОНЕЦ, САМЫЙ ГЛАВНЫЙ ВОПРОС. КАК ТЫ ОТНОСИШЬСЯ К САТАНИЗМУ?

АНДРЕЙ МАТВЕЕВ: Видимо, твой вопрос вызван тем, что символом FreeBSD является демон. Вопреки многочисленным заблуждениям, слово «daemon» не имеет ничего общего с воплощением зла и дословно означает «гений, дух-покровитель человека». В данном контексте «демон» можно рассматривать как независимое существо со своими собственными намерениями и желаниями. «Технически» демон в *nix представляет собой фоновый процесс, который выполняет всевозможные системные задачи. В Windows 2000/XP/2003 тоже есть демоны, просто Microsoft дала им другое, менее шокирующее имя — «сервисы». К слову, супер-сервер inetd раньше называли «супердемон» **C**

SPECIAL ОБЗОР

EASY

Доступный UNIX: Linux, FreeBSD, DragonFlyBSD, NetBSD, OpenBSD

СПб.: БХВ-Петербург, 2006 / Федорчук А.В. / 672 страницы
Разумная цена: 247 рублей

Казалось бы, о UNIX, Linux и BSD за последние годы написано множество книг, статей и сетевых материалов. Но многие — откровенная компиляция уже существующих книг и публикаций, либо что-то устаревшее и потерявшее актуальность. К тому же, львиная доля книг оставляет в тени других представителей семейства открытых POSIX-систем, ассоциируя Open Source исключительно с ОС Linux. В данной книге речь идет не о конкретной реализации, воплощенной в конкретном дистрибутиве или какой-либо BSD-системе, а о том, что все их объединяет. Изложены основные принципы, на которых базируются UNIX-подобные системы, даны приемы решения повседневных пользовательских задач. Отличный подарок для начинающих юниксоидов :).

MEDIUM

Windows и Linux на одном компьютере

М.: Лучшие книги, 2006 / Черникова С.В. / 208 страниц
Разумная цена: 93 рубля

Большинство пользователей зависает в пределах одной ОС из-за отсутствия желания менять комфортную среду, к которой уже притерся, на что-то новое и неизведанное, тем более, если это переход с Windows на Linux. И все потому, что мало кто знает о виртуальных компьютерах — программах, которые позволяют запускать большинство известных операционных систем как обычные программы Windows. Таким образом, чтобы сделать первые шаги за пределы Windows, не придется выходить из Windows! Другой вариант — установить Linux второй операционной системой, опять же, не удаляя Windows. Оба варианта подробно описаны в этой книге.

Если заинтересовался, можешь заказать любую книгу из обзора (по разумным ценам), не отрывая пятой точки от дивана или стула, в букинистическом интернет-магазине **«OS-книга»** (*www.osbook.ru*). Книги для обзора мы берем именно там.

HARD

Linux: Сетевая архитектура. Структура и реализация сетевых протоколов в ядре

СПб.: БХВ-Петербург, 2006 / Кенин А.М. / 464 страницы
Разумная цена: 172 рубля

Ядро Linux характеризуется стабильной и всесторонней реализацией семейства протоколов TCP/IP. Свободно доступный исходный код позволяет легко модифицировать и улучшать функциональные возможности экземпляров протоколов. Кроме того, улучшение возможностей ядра поддерживается принципом модулей ядра. Но прежде чем улучшать сетевые функциональные возможности Linux, необходимо разобраться с сетевой архитектурой Linux, что требует времени. Прибавь к этому скудную документацию по сетевой подсистеме Linux, и окажется, что эта книга необходима, если ты хочешь разобраться в процессах и структуре сетевой архитектуры Linux. Основополагающие PPP, IP, брандмауэры, маршрутизация, TCP, NAT, UDP и сокеты, из последних веяний — PPPoE в DSL, драйвер Bluetooth и т.п.

MEDIUM

Полный справочник по FreeBSD

М.: Издательский дом «Вильямс», 2005 / Родерик Смит / 672 страницы
Разумная цена: 327 рублей

Подробное руководство по инсталляции, конфигурированию и обслуживанию популярной операционной системы FreeBSD. Причем упор делается на администрирование FreeBSD, а не на обычное использование системы. Инсталляция системы, сосуществование с другими ОС, средства системного администрирования, управление процессами/разделами/файлами/учетными записями, инсталляция программного обеспечения, конфигурирование ядра, X Windows System, сетевое конфигурирование, брандмауэры, файловые серверы, почтовые серверы, web-серверы, офисные средства, графические средства, системная безопасность, поиск и устранение неполадок — обо всем этом достаточно подробно.

HARD

Linux: программирование в примерах

М.: КУДИЦ-ОБРАЗ, 2005 / Роббинс А. / 656 страниц
Разумная цена: 211 рублей

Пригодится не только тем, кто пишет под GNU/Linux, но и тем, кто пишет под различные системы Unix. Авторы все примеры сфокусировали на базовых API: управление памятью, файловый ввод/вывод, метаданные файлов, процессы и сигналы, пользователи и группы, поддержка программирования (сортировка, анализ и т.п.), интернационализация и отладка. Список короток, чтобы не пытаться научить всему и сразу. Просто после этой книги, посвященной основам, авторы планируют издать книги по межпроцессорному взаимодействию и сетям, по разработке программного обеспечения и переносимости кода, по многопоточному программированию и программированию графических интерфейсов пользователя (GUI). Если книга покажется достойной — жди продолжения.

MEDIUM

FreeBSD: установка, настройка, использование

СПб.: БХВ-Петербург, 2005 / Федорчук А.В. / 640 страниц
Разумная цена: 197 рублей

О существовании ОС FreeBSD слышали многие. Есть даже мнение, что это — один их самых удачных проектов на базе движения открытых исходных текстов. Во всяком случае, обычный эпитет для FreeBSD — «круто». Но мало кто из пользователей PC видел эту систему живьем. И еще меньше тех, кто использует ее в повседневной работе. Правда, одна из причин тому — очень скудная литература по FreeBSD именно на русском языке. Основной источник информации — переводы официальной документации проекта FreeBSD и устаревшие онлайн-описания. Эта книга — хорошая попытка прорвать информационную блокаду и доступно рассказать, как установить, настроить и использовать на практике FreeBSD.

SPECIAL ОПРОС

АЛЕКСЕЙ ПЕТРОВ

В IT 20 лет. Эксперт в области защиты данных, эксперт по компьютерным преступлениям, эксперт по сетевым коммуникациям и телефонии. Сертификаты от *Novell/3com/Bay/Siemens/Cisco/ISACA*. Консультант по вопросам ИТ-безопасности в Secproof Oy (*www.secproof.com*). Свободный консультант *Arhont.com*, *iPRO.lv*.

ВЛАДИМИР СЕЛЕЗНЕВ

Технический директор, интернет-провайдер «Синхролайн» (*www.sl.ru*).

АРТУР ЕНАЛИЕВ

Окончил МФТИ в 1999 году. На данный момент работает техническим директором ООО «Бест Хостинг».

АНТОН КАРПОВ

Специалист в области информационной безопасности. В «Х» пишет с переменной периодичностью вот уже несколько лет. Круг профессиональных интересов: сетевые атаки, безопасность UNIX-систем, безопасность беспроводных сетей..

КРИС КАСПЕРСКИ

Известен еще как мыщъх. Компьютеры грызет еще с тех времен, когда Правец-8Д считался крутой машиной, а дисковод с монитором были верхом мечтаний. Освоил кучу языков и операционных систем, из которых реально использует W2K, а любит FreeBSD 4.5. Живет в норе, окруженной по периметру компьютерами и стеллажами с литературой.

КОГДА И ГДЕ BSD НЕЗАМЕНИМА?

АЛЕКСАНДР АНТИПОВ: BSD отличается высокой надежностью и стабильностью работы, поэтому лучше всего использовать ее в многозадачных и критических системах — массовых почтовых системах, для организации публичных сервисов, хостинга и т.п.

АРТУР ЕНАЛИЕВ: Первое — это серверные системы высокой надежности, которые настраиваются один раз и работают долго и счастливо. Второе — касается лицензии BSD. Она позволяет на базе открытого ПО сделать «свой» продукт с закрытым кодом.

АНТОН КАРПОВ: В мире операционных систем общего назначения, коими являются BSD, Linux и Windows, действовало, действует и будет действовать главное правило: самая лучшая (для какой-либо задачи) ОС — та, которую лучше знаешь и которую лучше умеешь «готовить». Так что про незаменимость речи не идет: на Win можно построить маршрутизатор с фильтрацией пакетов и балансировкой нагрузки, а на BSD — контроллер домена Windows. Другое дело, что все операционки, конечно, разрабатываются с прицелом на конкретный, узко очерченный круг задач, которые они могут выполнить хорошо (в идеале — лучше всех).

FreeBSD разрабатывается с целью быть лучшей на x86-серверах общего назначения (почта, веб, базы данных и тому подобное). Усилия разработчиков направлены, в первую очередь, на оптимизацию для работы на мультипроцессорных (SMP) системах: планировщик задач (шедулер), библиотека нитей (threads), поддержка SMP в ядре — все пишется в угоду тому, чтобы FreeBSD была самой быстрой и производительной на многопроцессорных кластерах. Впрочем, и на однопроцессорных машинах тоже.

NetBSD главной своей целью имеет абсолютную портабельность — нет ни одной операционной системы, поддерживающей такое разнообразие аппаратных платформ. Достигается это как можно меньшим количеством платформозависимого кода в системе, использованием абстракций. Каждый новый релиз NetBSD выпускается для всех платформ (на данный момент их 57). Так что если задаться целью найти современную ОС, чтобы вдохнуть вторую жизнь в откопанный невесть где старенький компьютер экзотической архитектуры, то выбор будет однозначный — NetBSD.

OpenBSD не имеет аналогов в opensource-мире по двум параметрам: безопасность и сетевые возможности. Ребята из OpenBSD были первыми, кто включил в базовую систему такие механизмы проактивной безопасности, как защиту от переполнений буфера в собираемых их компилятором (модифицированным gcc) программах, защиту от выполнения кода в стеке (механизм W^X), рандомизацию адресов выделяемой памяти (модификации в malloc(3), mmap(2)). Они также внедрили механизмы privilege separation и privilege revocation во все сетевые сервисы и утилиты. Средство удаленного администрирования UNIX-машин, OpenSSH, являющееся стандартом де-факто в UNIX-мире, — также дело рук парней из OpenBSD. Причем все эти механизмы входят и включены в систему by default, в отличие от порочной практики, принятой в Linux-мире, согласно которой хорошую систему надо собирать, руководствуясь принципом «с миру по патчу».

Что касается сетевых возможностей, не побоюсь предположить, что «негласная» цель проекта OpenBSD — сделать, с точки зрения функционала, аналог популярных сетевых устройств от Cisco. В первую очередь речь здесь, конечно, идет о межсетевых экранах Cisco Pix. Но аналог открытый и свободный. OpenBSD «из коробки» умеет фильтровать и приоритезировать трафик с помощью мощнейшего пакетного фильтра pf, не имеющего аналогов. Настройка IPSec сводится к двум-трем телодвижениям, благодаря утилите ipsecctl(8), также не имеющей аналогов. То есть настройка IPSec стала не сложнее настройки правил файрвола. Есть поддержка агрегации сетевых интерфейсов (trunk) и прозрачного резервирования (failover) для построения отказоустойчивых кластеров, в том числе и для IPSec-соединений! В базовую систему входят демоны bgpd(8) и ospfd(8), для построения на базе OpenBSD динамического BGP или OSPF-маршрутизатора. Конечно, идеальных систем не бывает — разработчики OpenBSD не успевают следить за новым железом, да и производительность некоторых подсистем (файловая система, система нитей, поддержка SMP) оставляет желать лучшего. Однако, с точки зрения безопасности и по сетевым возможностям, OpenBSD — безусловно, система номер один, и не только в opensource-мире.

КРИС КАСПЕРСКИ: Рынок предлагает довольно большой выбор, а на цвет и вкус все фломастеры разные. BSD (особенно NetBSD) перенесена на множество платформ, от суперкомпьютеров до «контроллеров лифта» и прочих встраиваемых устройств. Все операционные системы семейства BSD бесплатны, поставляются в открытых исходных текстах (с правом модификации и доработки), хорошо масштабируются, выдерживают большую нагрузку даже на скромном железе, позволяя собрать приличный сервер, обслуживающий миллионы пользователей одновременно, оплатив только оборудование и, естественно, работу администратора. BSD неприхотлива, но для обращения с ней нужен хороший специалист, поскольку BSD ориентированна именно на специалистов.

АЛЕКСЕЙ ПЕТРОВ: BSD незаменима там, где максимально эффективно реализуются ее плюсы для решения конкретной задачи. Каждая операцион-

АЛЕКСАНДР АНТИПОВ

Руководитель проекта, автор/соавтор/корректор многочисленных статей ведущего отечественного портала по информационной безопасности *SecurityLab.ru*.

ная система имеет некие тактико-технические характеристики, обусловленные реализацией и внутренней организацией, а также поддержкой какого-то «железа». Но мало выбрать решение. Как правило, его надо реализовать и после этого еще и поддерживать. Для этого нужен знающий «механик»-администратор, который знает внутреннее устройство и может диагностировать неполадки и разобраться в их причине.

BSD будет незаменима там, где для «данной задачи» у нее будет больше баллов в сравнении с другими ОС. Скажем, в сравнении с Linux 2.2, BSD TCP/IP stack будет гораздо быстрее и на большой загрузке выдаст возможный максимум. TCP stack/netfilter BSD vs Linux 2.4 — уже в зависимости от ситуации и задачи придется выбирать либо BSD, либо Linux. BSD хуже справляется с некоторыми задачами (MySQL/SMP/threads/NUMA/SMP). Чего нельзя сказать о производительности некоторых сетевых апликаций. MySQL/SMP/Oracle/Java быстрее и лучше будут работать на Linux'e, nat/fw/ftpd/dns/apache — быстрее на BSD. Но во многих случаях с правильным тюнингом kernel'a эти тезисы очень спорны, и разрыв в производительности на одном и том же железе не так велик.

ВЛАДИМИР СЕЛЕЗНЕВ: BSD очень хороша для загруженных, «больших» хостинг-серверов, которые должны выдерживать нагрузку с большой посещаемостью, в сравнении с Linux. В BSD есть несколько очень удобных инструментов для хостинг-платформ, которых раньше не было в Linux, в частности, это простейшая (но надежная) система изоляции системных и прикладных приложений от основной машины (Jail — виртуальные серверы). В Linux бесплатные технологии виртуализации серверов появились не так давно. Очень давно используется система установки и обновления приложений (ports), — наверное, лучшая в своем роде.

ОТКРЫТЫЕ ИСХОДНИКИ: ПРОГРАММИРОВАНИЕ ИЛИ РЕЛИГИЯ?

АЛЕКСАНДР АНТИПОВ: И то, и другое. Долгое время открытые исходники были прежде всего религией, причем ортодоксальной. Такая религия привела к тому, что популярность этих ОС стала ничтожно мала по сравнению с Windows. Однако в последнее время, благодаря гигантам типа IBM и SUN и многомиллиардным вливаниям, у *Nix появился шанс отхватить долю рынка у Microsoft и Co. Правда, этому сильно мешают фанатично настроенные сторонники открытого кода.

АРТУР ЕНАЛИЕВ: С одной стороны, это искусство программирования, когда свой код не стыдно сделать открытым (показать другим). С другой стороны, образ мышления, можно сказать, даже религия, когда программист не просто решает поставленную задачу, а делает вклад в развитие огромного сообщества разработчиков ПО с открытым кодом, придавая, таким образом, своему занятию более глобальный смысл. Возможно, это и побуждает программистов делать открытый код более качественным и выверенным.

АНТОН КАРПОВ: Для кого как. Очевидно, что для харизматичного чудаковатого Столлмана открытые исходники — это уже давно религия, и сам он — апостол FSF ;). Для большинства же программистов, работающих над opensource-проектами, открытые исходники — это прежде всего возможность совместно заниматься общим делом. Принципиальное расхождение существует лишь в вопросе лицензирования открытого ПО. Как известно, самые популярные лицензии — это GPL (более открытая) и BSDL (более свободная). А вот вопрос использования лицензии, действительно, может легко перейти в религиозное русло :). Впрочем, для профессионального программиста вопрос качества ПО, конечно, гораздо важнее религиозных споров.

КРИС КАСПЕРСКИ: Это и программирование, и религия, причем довольно агрессивная. Открытый код позиционируется как универсальное решение всех проблем, сущий рай или, можно сказать, даже коммунизм. Закрытый код отметается сразу, даже если он работает лучше, быстрее, стабильнее. Для многих использование открытого софта является своеобразной формой протеста против Microsoft, и в этом есть свое рациональное зерно. Microsoft безраздельно властвует на рынке, навязывая нам свои уродские API, с не менее уродскими библиотеками, только потому, что большинство программистов даже не догадываются, что в этом мире кроме Windows есть что-то еще.

АЛЕКСЕЙ ПЕТРОВ: Для операционных систем открытый код гораздо более ценный критерий. Открытость кода позволяет правильнее и эффективнее писать апликации, понимая, что и как происходит внутри ОС. Всегда можно взять и проанализировать, что и как работает. Почти всегда можно взять базу кода и доработать его под себя, получив максимальный эффект. Не тратится время на изобретение колеса и велосипеда — значит, есть возможность двигаться дальше. Хороших алгоритмов и реализаций не так-то много, и патентование сильно тормозит и усложняет процесс развития в целом. Если кто-то запатентовал колесо — всем остальным на кубиках далеко не уехать, а многие текущие идеи и алгоритмы базируются на десятках лет опыта и вытекают из других — патентовать такие вещи в корне неправильно. Продажа «черных коробок», которые берут что-то на входе и выдают неизвестно что в результате — это часть бизнеса, защита идеи. Хорошо строить что-то из сложных «черных коробок» с сотней входов и выходов, без понятия логики — гораздо сложнее, часто конструкция может быть неустойчивой по абсолютно непонятным причинам. И я не против бизнеса и не ратую за то, чтобы ломать экономику, но часто выходит, что бизнес с радостью «за так» берет базу из BSD/GPL, но ничего туда не вкладывает!

ВЛАДИМИР СЕЛЕЗНЕВ: Открытые исходники — это очень удобно. Если у тебя что-то не работает в приложении или что-то работает, но, на твой взгляд, неправильно, то у тебя есть прекрасная возможность заглянуть внутрь программы и посмотреть, что конкретно происходит в этот момент. И, в конце концов, докопаться до сути проблемы, найти «баг» или узнать, в чем ошибся ты сам. Это, конечно, крайний вариант, и к нему редко обращаются, но это не позволяет опустить руки и сказать: «раз что-то не работает, то поделать с этим ничего нельзя». Всегда можно решить проблему.

ПОЧЕМУ В ОТКРЫТОМ LINUX БОЛЬШЕ ДЫР, ЧЕМ В ОТКРЫТОМ BSD?

АЛЕКСАНДР АНТИПОВ: Количество дыр — величина, зависящая от множества параметров: сложности кода, его длины, профессиональных навыков программистов и т.п. Главная причина в том, что BSD на протяжении множества лет разрабатывается строго определенной командой, а линукс прежде всего разрабатывается огромной армией фанатиков, профессиональный уровень многих из которых очень низок.

АНТОН КАРПОВ: Основных причин две. Первая — банальна: Linux просто популярнее BSD. Linux имеет поддержку больших компаний, что немаловажно для многих заказчиков. Вполне логично, что, чем более система распространена, тем более пристальное внимание ей уделяют эксперты по безопасности, да и просто взломщики. Конечно, это не означает, что если, скажем, OpenBSD войдет в каждый дом, то в ней обнаружат критические проблемы вроде RPC DCOM :). Однако факт остается фактом — чем популярнее и востребованнее система, тем чаще ее ковыряют эксперты по безопасности, и тем чаще в ней находятся проблемы безопасности, так как они — увы и ах! — есть везде.

Однако немаловажен и тот факт, что в Linux и *BSD-системах применяются разные модели разработки. FreeBSD разрабатывает узкий круг профессионалов — это люди, имеющие право внесения изменений в исходные коды системы (commit bit), каждый из которых ответственен за определенную подсистему (maintainer). В проекте также имеется офицер безопасности (security officer), следящий, в том числе, и за высоким уровнем качества программирования. Все важные патчи должны пройти через него, прежде чем будут добавлены в дерево исходных кодов. Что же касается OpenBSD, то здесь все очевидно — проект изначально имеет целью создание самой безопасной ОС на Земле, а, согласно лидеру проекта, Theo de Raadt, «безопасность определяется качеством» (под качеством понимается, конечно, и качество кода). И с этим трудно спорить. Разумеется, такой механизм более инертен, чем «базарная» модель разработки Linux-ядра, когда сотни разработчиков присылают патчи, за качеством кода которых порой никто не следит.

В результате мы имеем такую «вилку». Динамично развивающаяся ОС, подхватывающая поддержку всех новых технологий, но зато имеющая невысокое качество кода, что приводит к обнаружению все новых дыр. Или консервативная система, где на первое место ставится качество выпускаемого продукта, а не его функционал, что снижает вероятность наличия и, соответственно, обнаружения проблем безопасности.

КРИС КАСПЕРСКИ: Потому что «открытость» на безопасность практически никак не влияет. Аудит кода (особенно чужого) на предмет безопасности — весьма трудоемкое занятие. И наивно думать, что миллионы экспертов по всему миру не имеют никакого более интересного занятия, чем ковыряться в недрах Linux'а, который развивается весьма стремительно и объединяет как профессионалов, так и пионеров. Причем Linux — это фактически только ядро, разрабатываемое одной командой, с более или менее централизованной системой управления, а дистрибутивы клепают все, кто попало и из чего попало. Отсюда и дыры.

BSD развивается намного медленнее, причем базовый код, написанный еще черт знает когда, остается практически без изменений, что делает появление новых дыр достаточно маловероятным явлением (в OpenBSD за все восемь лет ее существования была обнаружена только одна серьезная дыра).

АЛЕКСЕЙ ПЕТРОВ: BSD писался и пишется инженерами — код перепроверяется и буквально вылизывается по крохам. В BSD хороший version control и менеджмент кода. Linux пишется разнородной группой энтузиастов-программистов, код часто просто не успевают проверять, темпы разработки ядра просто скоростные (особенно это относится к ядру 2.6.x).

ЧТО ВАЖНО ПРИ ОПТИМИЗАЦИИ СИСТЕМЫ?

АЛЕКСАНДР АНТИПОВ: Оптимизация состоит из двух этапов: оптимизация под аппаратную часть и оптимизация под программную среду. В первом случае большой эффект дает правильная компиляция ядра — включение необходимых параметров оптимизации при компилировании, компиляция под конкретно используемый тип процессора. Затем оптимизация сетевых настроек под тип используемой сетевой карты и особенности сетевой среды. В случае программной среды для каждого типа (почтовый сервер, web-сервер, база данных) можно писать большие статьи, но принцип остается неизменным — тюнинг параметров ядра, сетевого окружения и дисковой подсистемы.

АРТУР ЕНАЛИЕВ: Главное — не навредить. Другими словами, оптимизируя какой-либо параметр системы, важно следить за тем, чтобы другие параметры той же системы «не портились».

КРИС КАСПЕРСКИ: Важно знать, что ты делаешь. Вслепую много не наоптимизируешь. Прежде всего необходимо отбросить концепцию «бутылочного горлышка», то есть самого узкого места, тормозя-

щего все остальные. Если система настроена неправильно — тормозить будет все, хотя ярко выраженных «бутылочных горлышек» может и не быть, что сделает профилировщик бесполезной игрушкой и останется только эксперимент. Кстати говоря, влияние тех или иных параметров на производительность очень часто обнаруживается чисто случайно. Об этом не говорится в документации, и даже сами разработчики пожимают плечами, и только опыт...

АЛЕКСЕЙ ПЕТРОВ: Знание и понимание того, как и что работает. Изначально система «настроена» под некое среднее или завышено среднее, и оптимизация заключается в трех шагах. Первый — сбор и анализ статистики (скажем, каких процессов и операций производится больше, распределение и использование памяти, какие сетевые операции — продолжительность и особенности tcp/udp-сессий). Второй — выбор решения, в идеале — просчет решений и выбор более эффективного и оптимального (просто добавить память или поменять настройки ее распределения, конфиги squid/mysql/apache, плюс пересборка ядра). Третий — непосредственно реализация, оптимизация-тюнинг и подгонка решения (изменение стандартных значений tcp/ip-стека, сокеты, буферизация, timeout'ы...). Плюс первые два шага по новой.

ВЛАДИМИР СЕЛЕЗНЕВ: Оптимизировать можно сами приложения, которые ты используешь. Например, настроить MySQL, чтобы он использовал необходимое количество памяти или определенную схему работы с клиентами (Child vs Tread), в Apache можно отключать неиспользуемые модули, что уменьшает количество памяти, выделяемой для каждого клиента. Также можно перекомпилировать приложения под твои требования из исходных текстов, включив только то, что нужно, и исключив ненужные функции. Также желательно, чтобы приложение поддерживало работу на нескольких процессорах, если он у тебя не один. Есть замечательный документ «Getting Maximum Performance from MySQL», его положения часто можно перенести и на другие приложения. Эффект может достигать 30-50 процентов.

Второй вариант — когда ты оптимизируешь саму систему, и все приложения на ней начинают работать быстрее. В операционных системах на основе открытых исходных текстов можно перекомпилировать ядро системы. Что позволит максимально использовать аппаратные возможности машины. Здесь нужно четко указать, какое «железо» используется, и на основании этого будет собрано новое ядро, которое будет занимать меньше памяти и работать немного быстрее.

В общем, при оптимизации важно понимать, что для работы твоего приложения является «узким местом»: например, работа с диском, памятью, с процессором или с соединениями по сети. В каждом случае надо находить параметры, которые за это отвечают, и устранять проблему.

КТО ВЫЖИВЕТ: WIN, LINUX ИЛИ BSD?

АЛЕКСАНДР АНТИПОВ: Вопрос о выживании уже не стоит, все эти ОС на рынке более 10 лет. Этого достаточно, чтобы, как минимум, держаться на плаву. Вопрос в том, кто будет лидировать в ближайшее время, тоже не стоит — лидерство Windows очевидно и непоколебимо. Будут идти локальные войны за отдельные секторы рынка, позиции в которых Linux и FreeBSD традиционно сильны — массовые сервисы, хостинг, базы данных, распределенные вычисления и т.п.

АРТУР ЕНАЛИЕВ: Выживут все. Для всех этих систем на ближайшее время работы хватит.

АНТОН КАРПОВ: Как поклоннику BSD, конечно, хотелось бы видеть тотальный BSD World Domination. Но я не думаю, что в ближайшие годы из этих ОС кто-то должен обязательно умереть. И вот почему. Вокруг Linux, с одной стороны, уже давно нет того ажиотажа, что царил во времена 2.2 и 2.4 ядер. Не даром Линус Торвальдс был включен недавно CNN в десятку людей, больше не влияющих на информационную индустрию. Феерия по поводу «крутой и свободной» ОС прошла, и на первый план теперь выходят проблемы Linux — проблемы модели разработки, проблемы качества кода. Даже самые ярые поклонники этой ОС признают, что с разработкой ядра 2.6 ситуация близка к неразберихе: одни, вроде бы устоявшиеся, подсистемы выносят из ядра, другие ломают от релиза к релизу, про третьи между тем забывают (так, разработчики официально признали, что с подсистемой 802.11 в Linux — беда). Торвальдс время от времени делает заявления в духе «хватит патчей, все усилия направляем на стабильность».

С другой стороны, Linux имеет поддержку множества компаний, и многим пользователям наплевать, что там творится с ядром, пока такие крупные вендоры, как Red Hat или Suse выпускают свои релизы и продают техподдержку. Свободные BSD-системы также умирать не планируют, медленно, но верно прогрессируя. И пусть по некоторым показателям и функционалу они находятся там, где Linux был несколько лет назад, путь развития BSD кажется более продуманным и выверенным. BSD верит в эволюцию, а не в революцию. Пожалуй, главная проблема открытых BSD в том, что за ними не стоят крупные компании. Многие заказчики просто боятся доверять свой бизнес системе, не имеющей мощного коммерческого вендора, предлагающего не просто «коробку», а готовое решение.

Что же касается Windows, то проблемы безопасности в ней находили, находят и находить будут. Наличие в сетевой серверной ОС бреши вроде нашумевшей в 2004 году уязвимости в RPC DCOM, когда любой мог получить полный удаленный контроль над ОС, на которой даже не запущено ни одного сетевого сервиса, — уже достаточный повод для разработчиков, чтобы обильно посыпать себе голову пеплом, отправить операционку на свалку истории и забыть о ней, как о недоразумении. В Microsoft от-

лично понимают, что наличие компонентов IE в серверной системе — лишь дополнительная брешь в безопасности. Однако, чтобы исправить все накопившиеся годами ошибки, надо переписывать ОС «с нуля», а это почти нереальная задача.

КРИС КАСПЕРСКИ: Все три перечисленные системы существуют (или точнее даже сосуществуют) уже черт знает сколько лет, под них написано нефиговое количество софта, вложены нехилые деньги, обучены специалисты... Поэтому в обозримом будущем умирать никто не собирается. Но Windows, сосредоточенная в одних руках, имеет меньше шансов на выживание, чем Linux и BSD, которые никому не принадлежат, то есть принадлежат всем.

Лично я, увидев, что сделали с Windows 2000, долго плевался и сказал, что когда поддержка Windows 2000 будет прекращена, я лучше перейду на Debian или BSD, чем сяду за XP или, того хуже, Longhorn. Преимущество Linux/BSD в том, что они не навязывают своим пользователям никакой особенной идеологии. Это конструктор — что хочешь, то и собираешь. А вот попробуй запустить Windows без графического интерфейса, без IE и без кучи других не нужных мне вещей, причем так, чтобы работали нужные мне программы! В долговременной перспективе это означает лишь одно — BSD приобретает пользователей, а Windows их теряет.

ВЛАДИМИР СЕЛЕЗНЕВ: Выживут все, но в разных сегментах. Linux завоевывает свою долю на рынке серверов за счет других Unix-подобных операционных систем, наверное, в основном, с закрытым исходным кодом. Плюс небольшой процент насажденных десктоп-клиентов, например, решение на уровне руководства компании «пересадить» всех сотрудников на Linux. BSD — это почти полностью серверная платформа, отчасти она конкурирует здесь с Linux. А Windows никто реально не сможет потеснить с рынка десктопов в обозримом будущем, как не удалось это сделать MacOS и OS/2, не удастся и Linux.

ЧТО ПРОЩЕ СЛОМАТЬ: WIN, LINUX ИЛИ BSD?

АЛЕКСАНДР АНТИПОВ: Простота взлома определяется множеством параметров: слабой конфигурацией по умолчанию, наличием простых в эксплуатации незакрытых дыр, массовостью использования и т.п. Пару лет назад ответ на этот вопрос был бы очевиден, однако с выходом SP2 для Windows XP и будущим релизом Windows Vista явного лидера тут нет.

АРТУР ЕНАЛИЕВ: Покажите конфиги — скажу, что проще сломать.

КРИС КАСПЕРСКИ: Если все системы залатаны и сконфигурированы правильно, то один хрен их взломаешь, — нужно искать дыры, о которых никто не знает. И тут возникает противоречие: тексты Linux'а открыты и легко читаемы, но дыр там немного, а в OpenBSD, наверное, нет совсем). Тексты Windows закрыты (впрочем, их легко найти в осле), а процесс дизассемблирования отнимает намного больше сил и времени, чем анализ открытых кодов, но и дыр в Windows столько... Что, как говориться, чем больше их находишь, тем больше их остается.

АЛЕКСЕЙ ПЕТРОВ: «Проще» сломать то, что уже изначально слабее и изначально содержит в себе больше ошибок. Плюс еще один очень важный фактор — как это «что-то» настроено, насколько хорошо в этом вопросе разбирается администратор (оценивает риски, выбирает решение, настраивает, знает все нюансы, следит за работой системы, понимает механизмы и тонкости ее работы). Комбинация «дырявая ОС» + «хороший админ» — как правило, надежнее, чем «более секьюрная ОС» + «плохо разбирающийся/ленивый админ». Хотя идеал, естественно — «хороший админ» + «хорошая ОС».

BSD — хорошо выверяемый код, более долгая жизнь самого проекта и кода, больше количество пройденных проверок и неплохой уровень программирования (хотя порой бывают досадные логические ошибки в реализации, не связанные с безопасностью). Результат — неплохая безопасность в целом.

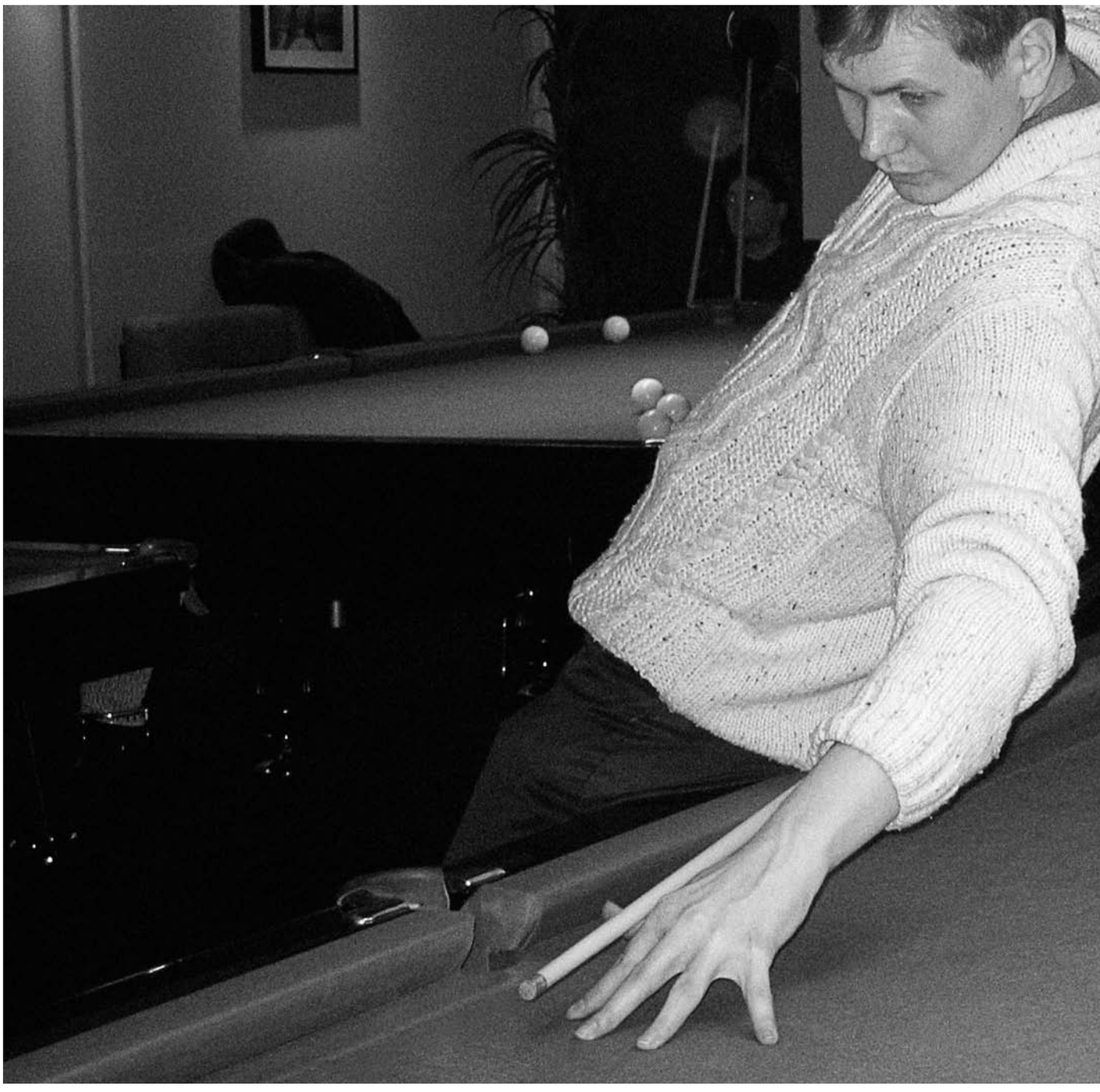
Windows — по количеству уязвимостей прочно держит первое место, ошибок там по-прежнему много на разных уровнях, но закрытость кода, его обилие, громадный объем, плюс запутанность скорее порождают нежелание без необходимости искать дыры. И вообще нужно следовать принципу: работает — лучше не трогать.

Linux'у хватает ошибок: отсутствие менеджмента и проверки кода, большая скорость разработки, огромный «штат» не очень хорошо организованной команды свободных приходящих-уходящих разработчиков. Громаднейшее количество «ползающих» по этому коду проверяющих — стабильное первое-второе место по количеству обнаруженных уязвимостей.

ВЛАДИМИР СЕЛЕЗНЕВ: Как показывает опыт, взломать можно все, но взлом происходит в 90% случаев на уровне приложения. Так что фактор ОС не так важен. Здесь важно, какие приложения ты используешь, как они поддерживаются, как часто и как быстро в случае проблем обновляются. И OpenSource-команда, и MicroSoft быстро реагируют и выпускают заплатки. Может быть, OpenSource иногда быстрее. Но, если команда разработчиков небольшая или приложение не популярное, заплатку можно ждать долго. С другой стороны, популярные приложения OpenSource чаще привлекают внимание хакеров.

Если ты силен в программировании, то систему с открытым кодом сломать проще. И в такой системе можно исправить ошибку в коде самому. OpenSource-программы написаны с меньшим количеством ошибок, работают надежнее, быстрее. Это мой выбор **C**

SPECIAL FAQ



На вопросы номера отвечает кернел-хакер, юникс-гуру, специалист по строительству и мелиорации, а также краснознаменный редактор братского журнала «Хакер» — **Андрей Матвеев**

Q ЧТО ДЕЛАТЬ? ПОМОГИТЕ СКОРЕЕ! С НЕДАВНЕГО ВРЕМЕНИ ЛОГИ МОЕЙ СИСТЕМЫ ЗАПОЛНЕНЫ ВОТ ТАКИМИ СООБЩЕНИЯМИ:

JUL 1 12:15:23 HOSTNAME SSHD[14196]: FAILED PASSWORD FOR INVALID USER ROOT FROM 208.195.218.73 PORT 51244 SSH2
JUL 1 12:15:24 HOSTNAME SSHD[19965]: RECEIVED DISCONNECT FROM 208.195.218.73: 11: BYE BYE

СУДЯ ПО ВСЕМУ, КТО-ТО ПЫТАЕТСЯ ПОДОБРАТЬ ПАРОЛИ К МОЕМУ ХОСТУ. ЧТО МОЖНО ПРОТИВОПОСТАВИТЬ ВЗЛОМЩИКУ?

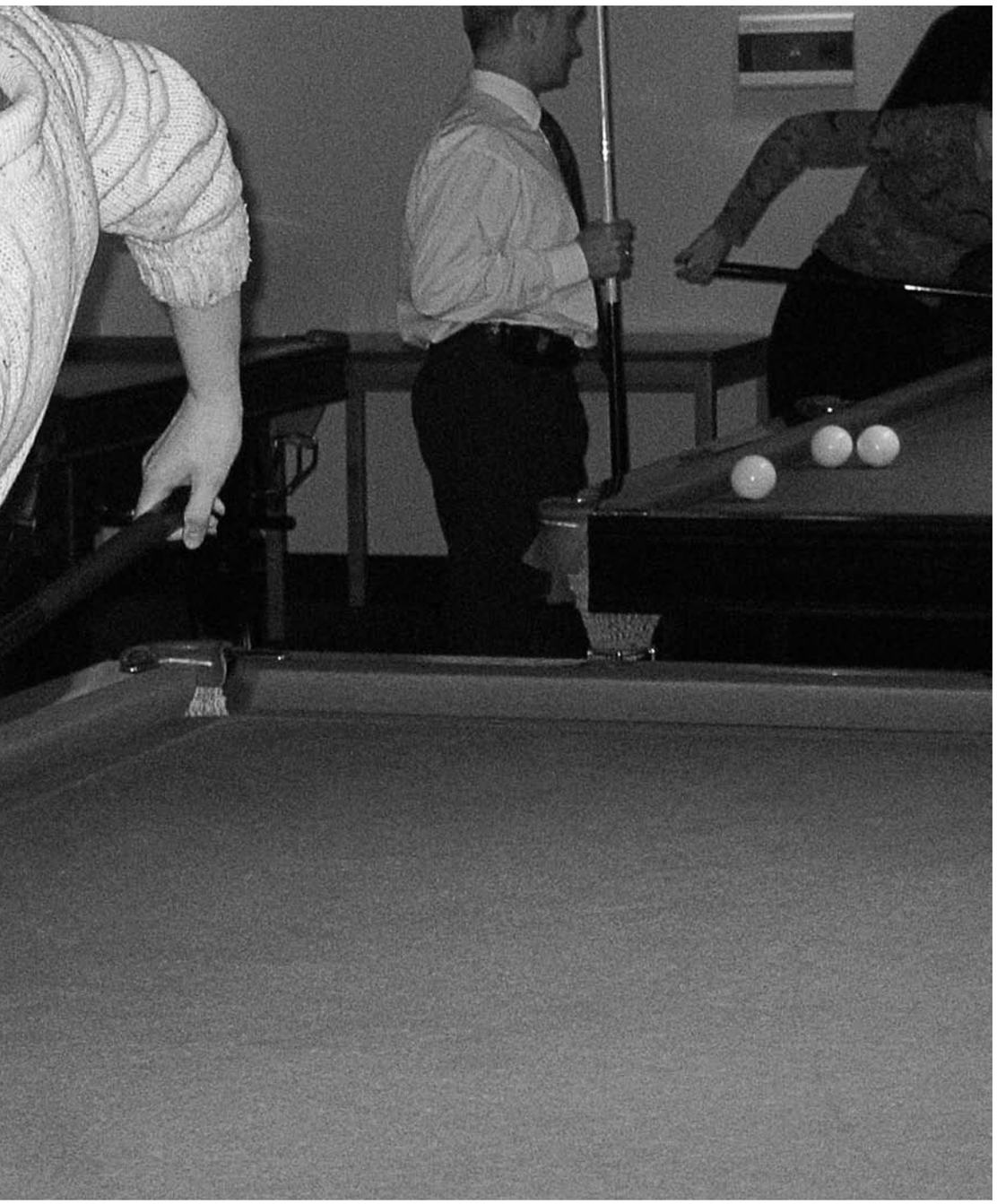
A Да, ты столкнулся с топорным видом атаки ssh bruteforce. Подавить чрезмерную активность переборщиков паролей можно, отслеживая максимальное количество подключений к конкретной службе. Приведу пример для фильтра пакетов pf:

vi /etc/pf.conf

```
table <sshbf> persist
block in log quick on $ext_if inet
from <sshbf>
pass in log on $ext_if inet proto tcp
to $ext_if port ssh keep state \
  (max-src-conn-rate 5/60, overload
<sshbf> flush global)
```

А теперь перезагружаем правила файрвола:

```
# pfctl -f /etc/pf.conf
```



При необходимости через определенные промежутки времени таблицу sshbf, с попавшими в нее IP-адресами злоумышленников, можно очищать:

```
# crontab -e
0 7 * * 6 /sbin/pfctl -t sshbf -T flush
2>/dev/null
```

В конфигурационном файле /etc/ssh/sshd_config следует определить список управления доступом:

 # vi /etc/ssh/sshd_config

```
PermitRootLogin no
PermitEmptyPasswords no
AllowUsers admin rdp vpn
```

Чтобы внесенные изменения вступили в силу, необходимо дать указание демону sshd перечитать свой конфиг:

```
# kill -HUP `cat /var/run/sshd.pid`
```

Q ХОЧУ ДРУЗЬЯМ ПОКАЗАТЬ СВОЙ РАБОЧИЙ СТОЛ. ПОДСКАЖИТЕ, КАК СДЕЛАТЬ СКРИНШОТ ВСЕГО ЭКРАНА X WINDOW?

A Для создания снимков экрана можно воспользоваться одной из следующих программ: gimp, xv, ksnapshot либо утилитой import из пакета ImageMagick:

```
$ import -display localhost:0.0 -window
root screenshot.jpg
```

Выполнив следующую команду, ты получишь графический дамп экрана:

```
$ xwd -root -out ~/screenshot
```

Однако подобный снимок удастся просмотреть только с помощью xwud:

```
$ xwud -in ~/screenshot
```

Напомню, что xwd и xwud являются штатными утилитами XFree86.

Q У МЕНЯ ЕСТЬ ЕЩЕ ДВА МАЛЕНЬКИХ ВОПРОСИКА. ОЧЕНЬ РАЗДРАЖАЕТ УЖАСНЫЙ ЗВУК В ТЕРМИНАЛКАХ XTERM, RXVT. КАК ОТ НЕГО ИЗБАВИТЬСЯ? И КАК ИЗМЕНИТЬ РАЗРЕШЕНИЕ БЕЗ ПЕРЕЗАПУСКА X-СЕРВЕРА?

A Желанная тишина достигается вот такой командой:

```
# xset b off
```

Для получения требуемого разрешения экрана запусти программу xrandr. Аргументом ключа '-r' выступает частота развертки.

```
# xrandr -s 1024x768 -r 85
```

Q ДЛЯ УДОБСТВА Я ХОТЕЛ БЫ НАБЛЮДАТЬ ЗА СОДЕРЖИМЫМ ВСЕХ ЛОГ-ФАЙЛОВ В ОДНОМ ТЕРМИНАЛЬНОМ ОКНЕ. ВОЗМОЖНО ЛИ ЭТО? И, ЕСЛИ ДА, ЧТО ДЛЯ ЭТОГО НУЖНО?

A Обрати внимание на программу screen из набора GNU-утилит. Вот так будет выглядеть ключевой момент твоего конфигурационного файла ~/.screenrc:

$ vi ~/.screenrc

```
screen -t logz1  1 tail -f /var/log/authlog
screen -t logz2  2 tail -f /var/log/daemon
screen -t logz3  3 tail -f /var/log/
maillog
screen -t logz4  4 tail -f /var/log/
messages
screen -t logz5  5 tail -f /var/log/xferlog
select 1
```

Кроме того, с помощью этой программы можно отсоединяться от консоли, а затем присоединяться обратно. Эта фича полезна, например, для запуска игровых серверов из стартовых скриптов системы:

 # vi /etc/rc.local

```
/usr/bin/su <username> -c "cd /usr/
local/games/quake3 && \
  /usr/local/bin/screen -d -m ./
q3ded +exec configfile.cfg"
```

ПОСТАВИЛ СЕБЕ FIREFOX. ВСЕ РАБОТАЕТ, ВОТ ТОЛЬКО СТРАНИЧКИ НЕ ОЧЕНЬ ШУСТРО ОТКРЫВАЮТСЯ! ПОСОВЕТУЙТЕ, КАК МОЖНО РАЗОГНАТЬ ОГНЕЛИСА?

Протокол HTTP 1.1 поддерживает множественные запросы — в сокет посылается сразу несколько запросов, а затем на них в строгом порядке ожидаются ответы. За счет такого подхода уменьшается количество сетевых пакетов и достигается существенный прирост скорости загрузки страниц.

Чтобы изменить настройки Firefox, в адресной строке следует набрать «about:config». При этом в новом табе браузера откроется редактор свойств не только самого Firefox, но и установленных в текущем профиле XPI-компонентов. Перечислю интересующие тебя опции (здесь Pipelining означает режим конвейерного соединения):

about:config

```
network.http.pipelining              ->
true
network.http.proxy.pipelining        ->
true
network.http.pipelining.maxrequests ->
попробуй значения 8, 16 и 32
nglayout.initialpaint.delay          -> 0
```

ХОЧУ, ЧТОБЫ МОЯ ФРЯХА ОДНОВРЕМЕННО ВОСПРОИЗВОДИЛА ЗВУК ОТ НЕСКОЛЬКИХ ПРИЛОЖЕНИЙ!

Попробуй создать несколько виртуальных звуковых каналов, как показано ниже:

```
# sysctl hw.snd.pcm0.vchans=4
# sysctl hw.snd.maxautovchans=4
```

А КАК ВЫЧИСЛИТЬ, СКОЛЬКО РАЗ В ЛОГЕ ВСТРЕЧАЮТСЯ, СКАЖЕМ, 3 ХОСТА, КОТОРЫЕ ЧАЩЕ ВСЕГО ПОСЕЩАЮТ МОЙ WEB-СЕРВЕР?

Хит-парад можно составить следующим образом:

```
# cat /var/www/logs/access_log | awk
'{print $1}' | sort | uniq -c | sort -r
-n | head -n 3
44783 192.168.2.2
1323 192.168.3.7
 360 192.168.3.10
```

Слева — количество посещений, справа — IP-адреса хостов, с которых эти посещения были произведены.

НА ДНЯХ ПРИКУПИЛ СЕБЕ НОВОМОДНЫЙ КПК. ПОДСКАЖИ, КАК МНЕ ПЕРЕКОДИРОВАТЬ НА НЕГО ПОНРАВИВШИЙСЯ ФИЛЬМ?

Да, ты не удержался, чтобы не похвастаться. Ok, записывай себе «на корочку» один действенный способ. Учти, что значение scale зависит от разрешения КПК и вычисляется по формуле: 220x? = Y/(X/220). В данном случае значение 220:165 приведено для разрешений 320x240 и 640x480.

```
$ mencoder berkova.avi -oac mp3lame -
ovc lavc -lavcopts \
 vcodec=mpeg4:vhq:vqmin=2:vqmax=20:
vmax_b_frames=2:vbitrate=100:vqcomp=0.6 \
 -vop scale=220:165,eq=15 -ofps 20 -
zoom -sws 2 -lameopts \
 cbr:br=32:aq=0:mode=3 -o berkova_pda.avi
```

ПОДСКАЖИТЕ, КАК АВТОМАТИЧЕСКИ УДАЛИТЬ В КАТАЛОГЕ /TMP ВСЕ ФАЙЛЫ С ИМЕНАМИ 'SESSION_*', КОТОРЫЕ БЫЛИ СОЗДАНЫ БОЛЕЕ ЧЕМ ТРИ ДНЯ НАЗАД?

Однако, уважаемый, не эту ли конструкцию ты ищешь:

```
# find /tmp -type f -name 'session_*' -
mtime +3 -print0 | xargs -0 rm -f
```

Предположу, что тебе также может понадобиться команда для удаления всех временных файлов нулевой длины:

```
# find /tmp -type f -name 'session_*' -
size 0 -print0 | xargs -0 rm -f
```

КАК БОРОТЬСЯ СО СМЕНОЙ IP-АДРЕСОВ БЕСПРОВОДНЫМИ КЛИЕНТАМИ ЛОКАЛЬНОЙ СЕТИ?

Мне очень нравится способ с привязкой IP к MAC с помощью bridge и pf. И совсем не важно, какие у нас клиенты — проводные или нет. В данном случае для создания моста достаточно одного внутреннего сетевого интерфейса ral0:

 # vi /etc/bridgename.bridge0

```
add ral0
blocknonip ral0
link0
-discover ral0
-learn ral0
flushall
// указываем MAC-адреса клиентских машин
static ral0 00:0f:ea:91:43:f6
static ral0 00:80:c8:2c:47:a1
up
// включаем фильтрацию
pass in on ral0 src 00:0f:ea:91:43:f6
tag user1
pass in on ral0 src 00:80:c8:2c:47:a1
tag user2
block in on ral0
```

Создаем и поднимаем псевдоустройство bridge:

```
# ifconfig bridge0 create
# sh /etc/netstart bridge0
```

Редактируем /etc/pf.conf:

 # vi /etc/pf.conf

```
// перечисляем используемые макросы
ext_if = "fxp0"
int_if = "ral0"

// указываем IP-адреса клиентских машин
user1 = "192.168.1.3"
user2 = "192.168.1.4"

// выпускаем валидных клиентов
в интернет
nat on $ext_if inet from { $user1,
$user2 } to any -> ($ext_if)

// немедленно блокируем всех остальных
block in log quick on $int_if from !
$user1 to any tagged user1
block in log quick on $int_if from !
$user2 to any tagged user2
```

Перезагружаем набор рулесетов файрвола:

```
# pfctl -f /etc/pf.conf
```

Я СЛЫШАЛ, ЧТО ПО УМОЛЧАНИЮ ВО FREEBSD ИСПОЛЬЗУЕТСЯ ШИФРОВАНИЕ ПАРОЛЕЙ МЕТОДОМ MD5. МОЖНО ЛИ ВКЛЮЧИТЬ БОЛЕЕ СЕКУРНЫЙ МЕХАНИЗМ?

Чтобы определить, какой метод шифрования используется в настоящий момент, достаточно заглянуть в файл /etc/master.passwd. Если пароли имеют префикс «1», то перед тобой MD5, если «$2a$», то — Blowfish. Скорее всего, у тебя применяется MD5. Чтобы перейти на Blowfish, измени одну строчку в файле /etc/login.conf:

 # vi /etc/login.conf

```
 :passwd_format=blf:\
```

И не забудь при этом обновить хэшированную базу данных:

```
# cap_mkdb /etc/login.conf
```

Стоит отметить, что в OpenBSD никаких телодвижений совершать не требуется. Blowfish задействован в этой ОС по умолчанию.

У МЕНЯ ЕСТЬ ШЕЛЛ НА ПОЧТОВИКЕ. СПИСОК ОТКРЫТЫХ ПОРТОВ НА ЭТОМ СЕРВЕРЕ: 22, 25 И 110. РАССКАЖИ, КАК МНЕ СОЗДАТЬ ШИФРОВАННЫЙ ТУННЕЛЬ, ЧТОБЫ НИКТО НЕ СМОГ ПЕРЕХВАТИТЬ МОИ ПАРОЛИ И КОРРЕСПОНДЕНЦИЮ.

В приведенном ниже примере на 10 минут будет создан ssh2-туннель для безопасного доступа к твоему POP3-серверу (127.0.0.1:8110 <--> mydomain.ru:110).

```
$ ssh -2 -4 -C -N -f -L 8110:localhost:
110 myname@mydomain.ru sleep 600
The authenticity of host 'mydomain.ru
(212.XX.XY.162)' can't be established.
RSA key fingerprint is f8:8a:da:f3:
6d:b4:dd:a8:3a:eb:30:8d:b6:be:e7:fe.
Are you sure you want to continue
connecting (yes/no)? yes
Warning: Permanently added 'mydomain.ru'
(RSA) to the list of known hosts.
```

Проверяем, забиндилась ли программа ssh на непривилегированный порт 8110:

```
$ netstat -na -f inet | grep 8110
tcp 0 0 127.0.0.1.8110     *.*      LISTEN
```

Теперь настраивай свой мылер на 127.0.0.1:8110 и спокойно забирай почту.

НА FTP.OPENBSD.ORG НЕ МОГУ НАЙТИ НИ ОДНОГО ISO-ОБРАЗА. ГДЕ ИХ МОЖНО ДОСТАТЬ?

Официальных iso-образов OpenBSD не существует — такова политика разработчиков. Но не стоит паниковать, давай сами сделаем исошку. Скачивай дистрибутивные файлы с **ftp://ftp.openbsd.org/pub/OpenBSD/3.9/i386/** в каталог /home/openbsd/image/3.9/i386 и выполняй следующие команды:

```
# cd /home/openbsd
# mkhybrid -b 3.9/i386/cdrom39.fs -c
boot.catalog \
 -lrvDJLN -hide boot.catalog -hide-
joliet boot.catalog \
 -V "obsd39" -o obsd39.iso image
```

Теперь, чтобы записать полученный образ в *BSD, можно воспользоваться программой cdrecord из пакета cdrtools:

```
# cdrecord -v dev=/dev/rcd0c obsd39.iso
```

А КАК РАЗМОНТИРОВАТЬ CDROM, ЕСЛИ Я НЕ УВЕРЕН, КАКАЯ ИМЕННО ПРОГРАММА СЕЙЧАС С НИМ РАБОТАЕТ?

Попробуй уничтожить все процессы, использующие /mnt/cdrom:

```
# cd /
# fuser -k -m /mnt/cdrom
# umount /mnt/cdrom
```

Также полезной может оказаться штатная утилита fstat, отображающая статус всех открытых файлов.

ДЕРЕВО ПОРТОВ FREEBSD ПРОСТО ОГРОМНОЕ. КАК МНЕ БЫСТРО НАЙТИ НУЖНУЮ ПРОГРАММУ И ПОСМОТРЕТЬ ВСЕ ЕЕ ЗАВИСИМОСТИ?

На сайте **www.freshports.org** можно воспользоваться поиском, введя в форму ключевые слова. Кроме того, по дереву портов можно произвести локальный серчинг:

```
$ cd /usr/ports
$ make search key=mutt
```

КАК МОЖНО МОНТИРОВАТЬ/РАЗМОНТИРОВАТЬ ISO-ОБРАЗЫ В FREEBSD?

Предположим, нам нужно смонтировать q3arena.iso в каталог /mnt/q3arena. Сначала подгрузим необходимый модуль ядра:

```
# kldload vn.ko
```

Теперь, для монтирования исошки, последовательно набирай:

```
# mkdir -p /mnt/q3arena
# vnconfig /dev/vn0c /home/username/
q3arena.iso
# mount -t cd9660 /dev/vn0c /mnt/q3arena
```

Обратная операция:

```
# cd /
# umount /mnt/q3arena
# vnconfig -u /dev/vn0c
```

В КОНФИГАХ FREE/NET/OPENBSD-ЯДЕР Я ВСТРЕЧАЛ КЛЮЧЕВОЕ СЛОВО MAXUSERS. ХОДЯТ СЛУХИ, ЧТО, УСТАНОВИВ ЭТОТ ПАРАМЕТР РАВНЫМ 512 ИЛИ 1024, МОЖНО СЕРЬЕЗНО ПОВЫСИТЬ БЫСТРОДЕЙСТВИЕ СИСТЕМЫ. ЭТО ПРАВДА?

С помощью maxusers задаются размеры некоторых внутренних таблиц ядра (максимальное число процессов, сетевых буферов и т.д.). Нет необходимости изменять умолчальное значение этой директивы, если только ты не являешься «счастливым» обладателем постоянно загруженного сервера.

ПОСОВЕТУЙ, ПОЖАЛУЙСТА, КАКОЕ-НИБУДЬ БЕЗЗЛОБНОЕ ЗАПАДЛОСТРОЕНИЕ. ХОЧЕТСЯ ИЗРЯДНО УДИВИТЬ ОДНОГО ЗНАКОМОГО АДМИНИСТРАТОРА.

Использование штатной утилиты logger открывает бескрайние просторы для полета фантазии. Предложу твоему вниманию маленький скрипт, который будет добавлять в файл /var/log/daemon фейковые записи об успешной работе POP3-сервера.

$ vi ~/fake.sh

```
#!/bin/sh

echo 'Authentication passed for pupkin'
| logger -i -t 'popa3d' -p daemon.info
echo '666 messages (31337 bytes)
loaded' | logger -i -t 'popa3d' -p
daemon.info
echo '666 (31337) deleted, 0 (0) left'
| logger -i -t 'popa3d' -p daemon.info
```

Скрипт сделан для удобства и наглядности, тот же самый функционал можно записать одной строчкой:

```
$ echo 'Authentication passed for pupkin\
n666 messages (31337 bytes) loaded\n666
(31337) deleted, 0 (0) left' | logger -
i -t 'popa3d' -p daemon.info
```

Не трудно представить себе лицо админа, обнаружившего в логах подобные строчки или записи типа: «bsd: no enough core», «no sane people allowed here, go home».

ПЫТАЮСЬ ПОДНЯТЬ WEB-ПОЧТУ. ЗАВЕЛ ПОДДОМЕН WEBMAIL.MYDOMAIN.RU, В АРАСНЕ НАСТРОИЛ ВИРТУАЛЬНЫЕ ДОМЕНЫ, ПРИКРУТИЛ PHP4, ПОСТАВИЛ SQUIRELLMAIL, ЗАПУСТИЛ. СОЕДИНЕНИЯ ПО HTTP ОБРАБАТЫВАЮТСЯ, А ПО HTTPS — НЕТ. Я ДАЖЕ ПЕРЕСОЗДАЛ ПРИВАТНЫЙ RSA-КЛЮЧ, НО ЭТО НЕ ПОМОГЛО. ЧТО ДЕЛАТЬ? КАК ЛЕЧИТЬ?

Проблема заключается в том, что проводить безопасные транзакции по протоколу

https при виртуальном хостинге на базе имен невозможно. Выход из ситуации — использование IP-aliasing'а. Приведу пример для OpenBSD (во Free и NetBSD настройка будет аналогична). Допустим, у нас есть сетевой интерфейс fxp0 с IP-адресом 192.168.1.1. Создадим для него IP-псевдоним 192.168.1.2 (примечание: для алиасов маска подсети всегда будет равна /32):

```
# ifconfig fxp0 inet alias 192.168.1.2
netmask 255.255.255.255
```

Теперь самое время поправить /etc/hostname.fxp0, чтобы после перезагрузки IP-псевдоним создавался автоматически:

 # vi /etc/hostname.fxp0

```
inet 192.168.1.1 255.255.255.0 NONE
inet alias 192.168.1.2 255.255.255.255
```

Далее приводим главный конфигурационный файл Apache в такой вид:

 # vi /var/www/conf/httpd.conf

```
// описываем наш виртуальный хост, в
качестве IP-адреса указываем IP-псевдоним
<VirtualHost 192.168.1.2:443>

// директория с дистрибутивом SquirellMail
 DocumentRoot /var/www/virtual/
webmail.mydomain.ru

// зарегистрированный поддомен и электрон-
ный адрес администратора почтовой системы
 ServerName webmail.mydomain.ru
 ServerAdmin admin@mydomain.ru

// относительные пути к журнальным записям
 ErrorLog logs/virtual.webmail.mydo-
main.ru-error_log
 CustomLog logs/virtual.webmail.mydo-
main.ru-access_log common

// создаем конфигурацию Apache SSL
 SSLEngine on
 SSLCertificateFile /etc/ssl/server.crt
 SSLCertificateKeyFile /etc/ssl/private/
server.key
 CustomLog logs/ssl_request_log \
   "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x
\"%r\" %b"
</VirtualHost>
```

И, чтобы изменения вступили в силу, набираем:

```
# apachectl configtest
# apachectl stop
# apachectl startssl
```

Я ЗАБЫЛ ПАРОЛЬ ROOT, КАК ЕГО ВОССТАНОВИТЬ ИЛИ ПОМЕНЯТЬ?

Интересно, ты его от своей или чужой системы забыл? :) Ну да ладно. Вот тебе универсальный способ. Загрузись в однопользовательском режиме, для этого в приглашении загрузчика введи «boot –s». Смонтируй командой «mount -u /» корневой раздел в режим чтения-записи. Затем с помощью «mount –a» примонтируй все остальные файловые системы (на самом деле, не все, а только те, что указаны в файле /etc/fstab без опции noauto). Все, теперь можно изменять пароль суперпользователя:

```
# passwd root
# shutdown -r now
```

КОГДА Я КОПИРУЮ ФАЙЛЫ ИЗ ПРИМОНТИРОВАННОГО FAT/NTFS РАЗДЕЛА В СВОЮ ДОМАШНЮЮ ДИРЕКТОРИЮ, ПРАВА ДОСТУПА АВТОМАТИЧЕСКИ ИЗМЕНЯЮТСЯ. ПОЯВЛЯЕТСЯ БИТ '+X'. МОЖНО ЛИ ОТ НЕГО ИЗБАВИТЬСЯ В АВТОМАТИЧЕСКОМ РЕЖИМЕ? А ТО, ЕСЛИ ДЕЛАТЬ ВРУЧНУЮ, ПОЛУЧИТСЯ ДОЛГО, ТАК КАК ФАЙЛОВ ОЧЕНЬ МНОГО.

Чтобы в текущей директории изменить права доступа к файлам на '-rw-r--r--' и вложенным подкаталогам на 'rwxr-xr-x', выполни эти две команды:

```
$ find . -type f -print0 | xargs -0
chmod 644
$ find . -type d -print0 | xargs -0
chmod 755
```

ЕСТЬ ШЛЮЗ НА БАЗЕ FREEBSD. ПОДСКАЖИ СПОСОБЫ ПОДСЧЕТА ТРАФИКА ДЛЯ КАЖДОГО ПОЛЬЗОВАТЕЛЯ ЛОКАЛЬНОЙ СЕТИ.

Для *BSD существует огромное количество различных считалок. Перечислю лишь отлично себя зарекомендовавшие: cnupm, ipa, ipacctd, ipcad, ipfm, trafd (я использую cnupm). Нельзя не упомянуть и ng_ipacct — подгружаемый модуль ядра, который работает через netgraph. Все вышеописанное ты без проблем найдешь в **www.google.ru** или **freshmeat.net**. Кроме того, подсчитать трафик можно с помощью средств, встроенных в штатные файрволы. Поскольку тебя интересует именно FreeBSD, приведу пример правила для ipfw:

```
ipfw add 100 count ip from any to
${local_ip_1} in
```

Полученные данные без особых проблем обрабатываются с помощью программы ipa либо самописного скрипта.

НА РАЗНЫХ СИСТЕМАХ (FREEBSD И OPENBSD) ВСТРЕЧАЛ СООБЩЕНИЯ ЯДРА ТИПА: «FILE: TABLE IS FULL» И «/VAR: OPTIMIZATION CHANGED FROM SPACE TO TIME». ЧТО ОНИ ОЗНАЧАЮТ?

Первая запись возвещает о том, что максимальное количество открытых файлов исчерпано. Чтобы посмотреть текущее значение и лимит, выполни команду:

```
$ pstat -T
8104/8192 open files
```

Переменная kern.maxfiles механизма sysctl(3) определяет максимальное число дескрипторов файлов. Каждый открытый файл, сокет или буфер использует дескриптор файла. Нагруженному серверу может понадобиться много тысяч дескрипторов файлов, в зависимости от количества одновременно выполняемых программ.

```
# sysctl -w kern.maxfiles=16384
kern.maxfiles: 8192 -> 16384
```

Во втором приведенном тобой сообщении ничего криминального нет. Ядро приняло решение об изменении алгоритма размещения файлов в файловой системе. В данном случае произошел переход с оптимизации по объему на оптимизацию по времени доступа. Более подробную информацию об этой оптимизации можно почерпнуть на страницах справочных руководств newfs(8) и tunefs(8).

КАК ПОЛУЧИТЬ ИСХОДНЫЙ КОД OPENBSD-ВЕТОК -STABLE И -CURRENT?

Первым делом укажи расположение общедоступного AnonCVS-сервера:

```
# export CVSROOT=anoncvs@anoncvs.ca.
openbsd.org:/cvs
```

Теперь для получения 3.9-STABLE:

```
# cd /usr
# cvs -fqz3 checkout -rOPENBSD_3_9 -P src
```

А вот так производится обновление до 3.9-STABLE (исходный код должен быть уже получен с помощью cvs checkout):

```
# cd /usr/src
# cvs -fqz3 update -rOPENBSD_3_9 -Pd
```

Чтобы обновить сырцы до ветки -CURRENT, указывать cvs'ный тэг не следует. Другими словами, нужно выполнить те же самые команды, только без '-rOPENBSD_3_9'.

ПО КАКОЙ-ТО НЕВЕДОМОЙ МНЕ ПРИЧИНЕ SENDMAIL НЕ ВИДИТ ЗАПИСЕЙ В /ETC/HOSTS! Я ПРОВОДИЛ ТЕСТИРОВАНИЕ НА РАЗНЫХ BSD-СИСТЕМАХ, НО ТАКОЕ ПОВЕДЕНИЕ ВЕЗДЕ ОДИНАКОВО. КАК ЗДЕСЬ БЫТЬ?

Все верно. По умолчанию Sendmail использует только службу DNS. Чтобы заставить его смотреть в /etc/hosts, создай однострочный файл /etc/mail/service.switch:

```
# vi /etc/mail/service.switch
```

После этого перезапусти демон командой:

```
# kill -HUP `head -1 /var/run/
sendmail.pid`
```

МОЖНО ЛИ ОБЕЗОПАСИТЬ СВОЙ СЕРВЕР, ПОМЕСТИВ СЕТЕВУЮ СЛУЖБУ В CHROOT?

Большинство сетевых демонов в *nix работает с правами суперпользователя. Если злоумышленник успешно проведет атаку, он получит возможность выполнять команды от имени root. Как все мы понимаем, ничего хорошего в этом нет. Для того чтобы обеспечить дополнительный уровень защиты и избежать возможного ущерба, следует запускать потенциально небезопасные демоны от имени непривилегированного пользователя в chroot'ной среде — среде с измененным для демона корневым каталогом (который на самом деле является обычным каталогом в файловой системе). Не имеет смысла запускать демон с правами суперпользователя в chroot, так как существуют пути, позволяющие руту выбраться из песочницы. Резюмируя вышесказанное, нельзя полностью обезопасить свой сервер, но можно значительно повысить его защищенность, если запускать сетевые службы в chroot-окружении.

ТЕПЕРЬ ПОНЯТНО. А ЧТО ИМЕННО НУЖНО СДЕЛАТЬ, ЧТОБЫ ПОСАДИТЬ, НАПРИМЕР, APACHE В CHROOT? ОПИШИ ХОТЯ БЫ В ДВУХ СЛОВАХ.

Для запуска Apache в среде chroot нужно создать миниатюрную копию системного дерева подкаталогов и поместить туда все, что необходимо для работы сервера, включая специальные файлы устройств и загружаемые библиотеки. Во всех конфигурационных файлах подопытного индейца следует указывать абсолютный путь относительно chroot'ного каталога. После выполнения этих действий httpd будет замыкать себя в директории /var/www, сбрасывать привилегии до пользователя www или nobody, а затем уже стартовать для принятия запросов.

ПРИ УСТАНОВКЕ OPENBSD Я НЕ РАССЧИТАЛ РАЗМЕР СВОПА. SQUID И MYSQL ЗАХВАТЫВАЮТ ВСЮ ДОСТУПНУЮ ОПЕРАТИВКУ И СВОП! ОБНОВИЛСЯ ДО ПОСЛЕДНИХ ВЕРСИЙ, НО ЭТО НИЧЕГО НЕ ДАЛО. КАК С МИНИМАЛЬНЫМИ ЗАТРАТАМИ СПРАВИТЬСЯ С ЭТОЙ ПРОБЛЕМОЙ? А ТО ПАМЯТЬ НЕ ОЧЕНЬ ХОЧЕТСЯ ДОКУПАТЬ!

Наверное, рекомендация здесь может быть только одна: добавить второй swap. Сделать это можно следующим образом (в примере объем свопа составляет 256 Mb):

```
# dd if=/dev/zero of=/home/swap
bs=1k count=262144
# chmod 600 /home/swap
# swapctl -a /home/swap
```

Посмотреть список активных файлов подкачки можно так:

```
$ swapctl -l
Device  1K-blocks   Used   Avail  Capacity
Priority
swap_device    524160  0  524160    0%    0
/home/swap     262144  0  262144    0%    0
Total    786304  0  786304 0%
```

Чтобы дополнительный своп автоматически монтировался при загрузке системы, добавь в конец файла /etc/fstab следующую строчку:

```
# vi /etc/fstab
/home/swap /home/swap swap sw 0 0
```

Напомню, что файл подкачки должен находиться в разделе, смонтированном без включенного механизма Soft Updates.

ТЫ УПОМЯНУЛ ПРО SOFT UPDATES. Я МНОГО РАЗ СЛЫШАЛ ЭТОТ ТЕРМИН. ЧТО ОН ОЗНАЧАЕТ?

Это механизм мягких обновлений, при котором упорядоченные операции записи выполняются без участия журнального файла (это можно назвать своеобразным кэшированием). Таким образом, существенно увеличивается скорость создания и удаления файлов. Я обычно включаю Soft Updates для всех файловых систем, кроме тех, что смонтированы к корневому разделу и /cvs.

А ЧТО ТАКОЕ SYSCALLS?

Syscalls — системные вызовы — низкоуровневые обращения непосредственно к ядру операционной системы для выполнения определенной функции. Как правило, во время написания программы используются вызовы функций из системной библиотеки libc, которые в процессе выполнения этой программы транслируются в syscalls. Кстати, ничто не мешает тебе кодить системные вызовы напрямую. Подними подшивку Хакера и СпецХакера: мыщъх неоднократно рассматривал эту тему.

НЕ ПОДКИНЕШЬ СПИСОК ПРОГРАММ ДЛЯ ВЗЛОМА БЕСПРОВОДНЫХ СЕТЕЙ?

Держи карман шире: kismet (с festival и gpsd), tcpdump, ethereal, aircrack (состоит из airodump для сбора пакетов, aireplay для внедрения пакетов в сеть, aircrack для непосредственного взлома ключа и airdecap для расшифровки WEP/WPA дампов), а также void11. Этого набора для первых вардрайверских шагов будет вполне достаточно.

МОЖНО ЛИ ОБМАНУТЬ СКАНЕР NMAP И КАК ЗАБЛОКИРОВАТЬ СКАНИРУЮЩИХ?

Предложу тебе на выбор два способа: с помощью pf и связки portsentry+ipfw, а ты уже сам выберешь подходящий тебе вариант.

Для packet filter:

 # vi /etc/pf.conf

```
// обманываем nmap
scrub in

// блокируем и регистрируем попытки
сканирования
block in log quick on $ext_if inet
proto tcp all flags FUP/FUP
block in log quick on $ext_if from
any os NMAP
```

Для portsentry и ipfw:

 # vi /usr/local/psionic/portsentry/portsentry.conf

```
TCP_PORTS="42,88,135,139,145,389,443,445,
464,593,636,637,1025,1026,1027,1029,1433
,3372,3389"
UDP_PORTS=""
IGNORE_FILE="/usr/local/psionic/
portsentry/portsentry.ignore"
RESOLVE_HOST = "0"
BLOCK_TCP="1"
KILL_ROUTE="/sbin/ipfw add 1 deny all
from $TARGET$:255.255.255.255 to any"
KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"
PORT_BANNER="STAY OFF MY COMPUTER!"
```

Вот и все, надеюсь я ответил на все твои вопросы

hard

маленький, но вместительный

ТЕСТИРУЕМ FUJITSU MHV2160BT

КОСТРОВ АНДРЕЙ

технические характеристики:

НАЗВАНИЕ:	**FUJITSU MHV2160BT**
ОБЪЕМ:	**160 Гб**
ИНТЕРФЕЙС:	**SATA 150**
СКОРОСТЬ ВРАЩЕНИЯ:	**4200 об/мин**
ОБЪЕМ КЭШ-ПАМЯТИ:	**8 Мб**
КОЛИЧЕСТВО ДИСКОВ:	**3**
КОЛИЧЕСТВО ГОЛОВОК:	**6**
ПОДДЕРЖКА NCQ:	**есть**
РАЗМЕРЫ:	**70x100x12,5 мм**
МАССА:	**0,135 кг**
ЦЕНА:	**около $200**

Несмотря на то, что самая распространенная скорость вращения магнитных пластин для 2,5” винчестеров равна 5400 оборотов в минуту, а отдельные разработчики изготавливают 7200 оборотистых моделей, на рынке еще можно встретить жесткий диск, у которого магнитные диски вращаются со скоростью 4200 оборотов в минуту. Накопитель FUJITSU MHV2160BT объемом 160 Гб как раз относится к семейству четырех тысячников и, кроме скорости вращения пластин, он выделяется большой толщиной, что объясняется тем, что внутри его корпуса находятся три магнитных диска и шесть головок, в то время как у большинства 2,5 винчестеров используется не более двух магнитных пластин. Накопитель оснащен кэш-памятью емкостью 8 Мб, работает по интерфейсу SATA с максимальной пропускной способностью 150 Мб в секунду и поддерживает технологию NCQ (Native Command Queuing).

➔ **результаты тестирования основных физических параметров накопителя.** Испытание проводилось при помощи двух программ: пиковая скорость интерфейса и время случайного доступа измерялись утилитой HD Tach, а скорости линейного и случайного чтения — посредством плагина Disk Benchmark популярной программы AIDA32.

ПИКОВАЯ СКОРОСТЬ ИНТЕРФЕЙСА:	**115,4 Мб/с**
ВРЕМЯ СЛУЧАЙНОГО ДОСТУПА:	**19,6 ms**
СКОРОСТЬ ПОСЛЕДОВАТЕЛЬНОГО ЧТЕНИЯ (МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ):	**31 Мб/с**
СКОРОСТЬ ПОСЛЕДОВАТЕЛЬНОГО ЧТЕНИЯ (СРЕДНЕЕ ЗНАЧЕНИЕ):	**24,5 Мб/с**
СКОРОСТЬ ПОСЛЕДОВАТЕЛЬНОГО ЧТЕНИЯ (МИНИМАЛЬНОЕ ЗНАЧЕНИЕ):	**15,2 Мб/с**
СКОРОСТЬ СЛУЧАЙНОГО ЧТЕНИЯ (МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ):	**31,7 Мб/с**
СКОРОСТЬ СЛУЧАЙНОГО ЧТЕНИЯ (СРЕДНЕЕ ЗНАЧЕНИЕ):	**24,9 Мб/с**
СКОРОСТЬ СЛУЧАЙНОГО ЧТЕНИЯ (МИНИМАЛЬНОЕ ЗНАЧЕНИЕ):	**14,9 Мб/с**

В тестах, измеряющих скорость последовательного и случайного чтения, накопитель показал прекрасную производительность, совсем немного уступив винчестерам со скоростью вращения магнитных пластин 5400 оборотов в минуту и опередив большинство 4200 оборотистых жестких дисков. Но результат теста максимальной скорости чтения из буфера не слишком высок — всего 115,4 Мб в секунду, что не раскрывает потенциал интерфейса SATA 150. Но время случайного доступа откровенно разочаровывает.

➔ **результаты тестирования при помощи пакета PCMark05.** Чтобы оценить производительность жесткого диска в операциях максимально приближенных к задачам, которые выполняет накопитель при повседневном использовании, мы применили пять дисковых тестов из популярного пакета PCMark05.

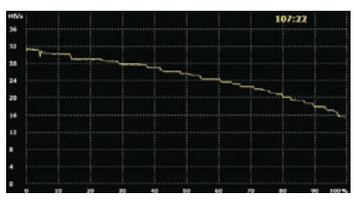
ТЕСТ XP STARTUP:	**5,18 Мб/с**
ТЕСТ APPLICATION LOADING:	**4,58 Мб/с**
ТЕСТ GENERAL USAGE:	**3,78 Мб/с**
ТЕСТ VIRUS SCAN:	**66,35 Мб/с**
ТЕСТ FILE WRITE:	**29,22 Мб/с**

Жесткий диск продемонстрировал высокую производительность для накопителя с невысокой скоростью вращения магнитных пластин. Быстродействие оказалось на уровне 5400 оборотистых винчестеров.

➔ **максимальная температура и акустический шум.** На протяжении всего тестирования отслеживалась температура при помощи программы DTemp. Оценка акустического шума, издаваемого устройством, производилась при отключенном вентиляторе блока питания для различных режимов работы жесткого диска. Невысокая скорость вращения магнитных дисков самым положительным образом сказалась на температуре, которая не превысила 35 градусов. А шум можно было услышать лишь при активной работе накопителя.

FUJITSU MHV2160BT можно охарактеризовать как очень быстрый накопитель относительно моделей со скоростью вращения дисков 4200 оборотов в минуту, слабо греющийся и практически бесшумный в работе, но следует обратить внимание на его большую толщину относительно 2,5 дюймовых винчестеров **C**

Очень высокая производительность при операции последовательного чтения для накопителя со скоростью вращения магнитных дисков 4200 оборотов в минуту.



Test_lab выражает благодарность за предоставленное на тестирование оборудование компании ПИРИТ: (495) 974-3210, **www.pirit.ru**

сэкономь деньги — закажи журнал в редакции

ВЫГОДА

Цена подписки до 15% ниже, чем в розничной продаже
Бонусы, призы и подарки для подписчиков
Доставка за счет редакции

ГАРАНТИЯ

Ты гарантированно получишь все номера журнала
Единая цена по всей России

СЕРВИС

Заказ удобно оплатить через любое отделение банка
доставка осуществляется заказной бандеролью или курьером

КАК ОФОРМИТЬ ЗАКАЗ

1 Заполнить купон и квитанцию
2 Перечислить стоимость подписки через любой банк.
3 Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:

— по электронной почте: **subscribe@glc.ru**;
— по факсу: (495) 780-88-24;
— по адресу: 119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

Внимание!
Подписка оформляется в день обработки купона и квитанции.

— купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней.
— купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.

Рекомендуем использовать электронную почту или факс. Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можно оформить с ноября.

ПОДПИСКА ДЛЯ ЮРИДИЧЕСКИХ ЛИЦ

Москва: ООО «ИНТЕР-ПОЧТА» (495) 500-00-60 www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.

подписной купон

стоимость заказа на Хакер Спец + CD

6 месяцев — 900 руб. 00 коп. | **12** месяцев — 1740 руб. 00 коп.

стоимость заказа на комплект Хакер Спец + Хакер + Железо

6 месяцев — 2550 руб. 00 коп. | **12** месяцев — 5040 руб. 00 коп.

прошу оформить подписку:
☐ на журнал Хакер Спец + CD
☐ на комплект Хакер Спец + Хакер + Железо
на месяцев
начиная с 200 г.
☐ Доставлять журнал по почте на домашний адрес
☐ Доставлять журнал курьером на адрес офиса (по г. Москве)
Подробнее о курьерской доставке читайте ниже*
(отметьте квадрат выбранного варианта подписки)
Ф.И.О.
дата рождения
адрес доставки:
индекс
область/край
город
улица
дом корпус
квартира/офис
телефон ( )
e-mail
сумма оплаты

*Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «_____» с _____ 200_ г. (месяц)	

Ф.И.О.
Подпись плательщика
Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «_____» с _____ 200_ г. (месяц)	

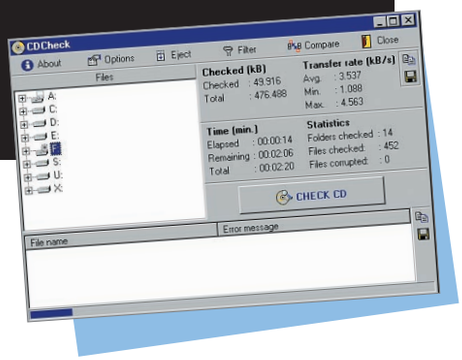
Ф.И.О.
Подпись плательщика
Кассир

ПО ВСЕМ ВОПРОСАМ, СВЯЗАННЫМ С ПОДПИСКОЙ, ЗВОНИТЕ ПО БЕСПЛАТНЫМ ТЕЛЕФОНАМ: **780-88-29** (ДЛЯ МОСКВИЧЕЙ) И **8-800-200-3-999** (ДЛЯ РЕГИОНОВ И АБОНЕНТОВ МТС, БИЛАЙН, МЕГАФОН). ВСЕ ВОПРОСЫ ПО ПОДПИСКЕ МОЖНО ПРИСЫЛАТЬ НА АДРЕС: **info@glc.ru**

Soft

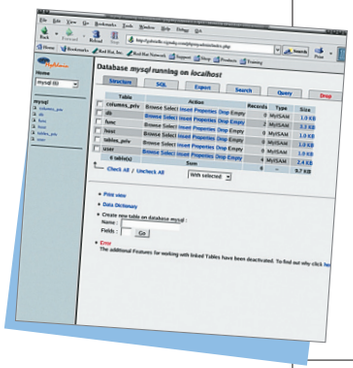
noname

НАИСВЕЖАЙШИЕ ПРОГРАММЫ ОТ NNM.RU
DOC@NNM.RU



CDCheck 3.1.12.0

Новая версия простой в использовании утилиты CDCheck. CDCheck помогает обнаружить повреждения сменных носителей и восстановить находящиеся на них данные. Ты можешь проверить компакт-диски, дискеты, предназначенные для ZIP и трехдюймовых дисководов. В принципе, можно проверить любой накопитель, при условии, что он доступен средствам операционной системы (виден в проводнике Windows). CDCheck сканирует диск и точно определяет местоположение дефектных участков. Причины повреждений могут быть самыми разными, но, вне зависимости от их происхождения, программа помогает вовремя заметить опасность и, возможно, спасти данные. В качестве дополнительной меры программа может выполнить сравнение данных с их образцовой копией в папке, расположенной на жестком диске (или другом накопителе). CDCheck сверяет содержимое файлов, выявляя все расхождения. Кроме того, программа может сформировать файлы контрольных сумм, дающие дополнительный признак того, что файлы на сменном диске не изменились. CDCheck имеет интерфейс на нескольких языках, включая русский.



phpMyAdmin 2.8.2 Final

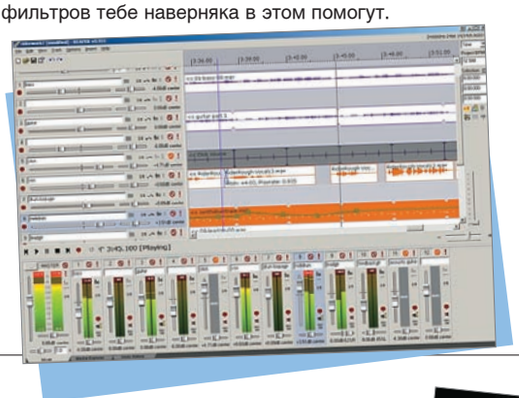
PhpMyAdmin 2.8.2: Web-мастерам посвящается! phpMyAdmin 2.8.2 — специальная программа, написанная на PHP, которая предназначена для администрирования и управления MySQL-серверами через сеть. Программа бесплатна и имеет русский интерфейс.

Google Video Player 1.0.1.0 Beta

Гугл предоставляет очень полезные сервисы. Например, на **video.google.com** каждый желающий вправе разместить видеоданные. Трабла в том, что гугл хранит видео в хитроумном формате. Но проиграть его всегда можно с помощью этого проигрывателя.

REAPER 0.977

Это бесплатное приложение позволит тебе записывать, редактировать и рендерить музыкальные композиции, состоящие из нескольких треков. Тьма всевозможных функций и фильтров тебе наверняка в этом помогут.



WinRAR Version 3.60 beta 6

Вышла новая бета WinRar — одного из самых известных архиваторов. То, что он поддерживает архивацию в формате RAR, это, вероятно, объяснять не надо. Кроме того, программа умеет работать с архивами ZIP, CAB, ARJ, LZH, TAR, GZ, ACE 2.0, BZIP, JAR, UUE, GZIP, BZIP2 и 7-Zip. При этом она обладает многочисленными полезными возможностями: шифрованием, поддержкой непрерывных (solid) архивов, в которых степень сжатия может быть на 10 — 50% больше, чем при обычных методах сжатия, специальным алгоритмом для сжатия мультимедийных файлов, поддержкой многотомных архивов и еще многим другим.

WinBackup Pro v2.20

Программа предназначена для автоматического создания резервных копий указанных файлов и папок по заданному расписанию. Ты можешь поручить программе автоматически (без твоего участия) сохранять самые важные и часто используемые файлы (документы, базы данных (например, 1С), бухгалтерские файлы, электронные письма, фото и т.д.). Это, безусловно, будет полезно тем, кто хочет быть абсолютно уверен в сохранности своих данных.

Photo Collage v1.40

Photo Collage — это полнофункциональная утилита для создания комбинированных коллажей. Кроме того, этот графический инструмент может создавать обложки для CD и DVD, обои для рабочего стола и всякую web-графику. В настройках программы много разных шаблонов, масок, рамок, что позволяет работать быстро и без напряга. Photo Collage может выполнять и функции просмотрщика изображений с возможностью скрытия своих приватных папок от посторонних глаз.

Catalyst 6.6

Июньская сборка драйверов от ATI для видеокарт Radeon. Данная сборка поддерживает: X1900, X1800, X1600, X1300, X850, X800, X700, X600, X550, X300, 9x00. В Catalyst 6.6 входит:

- RADEON DISPLAY DRIVER 8.263;
- MULTIMEDIA CENTER 9.14;
- HYDRAVISION 3.25.0006;
- HYDRAVISION BASIC EDITION 3.25.9006;
- REMOTE WONDER 3.03;
- WDM DRIVER INSTALL BUNDLE;
- SOUTHBRIDGE/IXP DRIVER;
- CATALYST CONTROL CENTER 6.6.

jetAudio v. 6.2.6.8330 Plus VX

JetAudio — один из самых лучших мультимедиа-центров («все в одном») с большими возможностями. Вот некоторые из них:

- ПРОИГРЫВАНИЕ МУЗЫКАЛЬНЫХ И ВИДЕОФАЙЛОВ ВСЕХ ПОПУЛЯРНЫХ ФОРМАТОВ (MP3, MP2, WAV, MID, REAL AUDIO/VIDEO, S3M, MOD, MPG, AVI, MOV, VIDEO/AUDIO CD, REALPLAYER G2 И ДР.);
- ЗАПИСЬ CD-R И CD-RW ДИСКОВ В АУДИОФОРМАТЕ;
- СВОБОДНОЕ РЕДАКТИРОВАНИЕ ТЭГОВ (ПОДДЕРЖКА ВНЕШНИХ И ЛОКАЛЬНЫХ БАЗ CDDB);
- ОЦИФРОВКА АУДИОДИСКОВ;
- КОНВЕРТИРОВАНИЕ АУДИО- И ВИДЕОФАЙЛОВ ИЗ ОДНОГО ФОРМАТА В ДРУГОЙ;
- ЗАПИСЬ ЗВУКА С ЛЮБЫХ ИСТОЧНИКОВ;
- ПРОСЛУШИВАНИЕ ИНТЕРНЕТ-РАДИО;
- ОЧЕНЬ УДОБНАЯ КАТАЛОГИЗАЦИЯ ФАЙЛОВ МУЛЬТИМЕДИА;
- СПОСОБНОСТЬ ПРИМЕНЯТЬ ВСТРОЕННЫЕ 3D-ЗВУКОВЫЕ ФИЛЬТРЫ К ЛЮБОМУ АУДИОИСТОЧНИКУ (ЕСТЬ ФИКСИРОВАННЫЕ НАСТРОЙКИ ROOM, HALL, STAGE, STADIUM). КРОМЕ ЭТОГО, У ПРОГРАММЫ ИМЕЮТСЯ МНОГОПОЛОСНЫЙ ЭКВАЛАЙЗЕР И РЕГУЛЯТОРЫ REVERB И 3D, ПОЗВОЛЯЮЩИЕ ДОБИТЬСЯ НАИЛУЧШЕГО КАЧЕСТВА ЗВУКА.
- ПОДДЕРЖКА СКИНОВ И ВИЗУАЛИЗАЦИИ И МНОГОЕ ДРУГОЕ...

Fresh UI 7.62

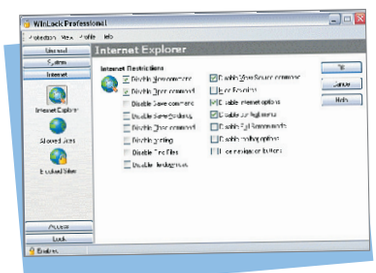
Это утилита для тонкой настройки Windows — подстройки интерфейса под свой вкус, оптимизации системы, включая ее «железные» установки, выбора системной политики в отношении пользователей и т.п.

WinLock 4.45 Professional

WinLock — программа является законченным решением для обеспечения безопасности компьютеров, работающих под управлением операционных систем Windows 95/98/ME или Windows NT/2000/XP. Независимо от того, находится ли компьютер в личном или в общем пользовании, WinLock гарантирует, что только авторизованные пользователи смогут получить доступ к важной информации.

Также WinLock позволяет запретить использование комбинаций клавиш Windows (таких как Alt-Ctrl-Del, Alt-Tab, Ctrl-Esc и так далее), блокировать рабочий стол Windows, настраивать меню «Пуск», скрывать кнопку «Пуск» и «Панель задач» и многое другое.

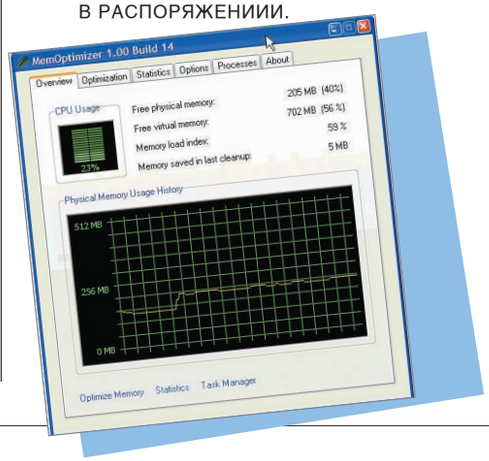
Функции «Блокировать окна» и «Блокировать файлы» позволяют блокировать приложения, окна «Проводника» Windows («Мой компьютер», «Корзина» и прочие) и выбранные файлы. Используя программное решение WinLock, ты можешь не беспокоиться о том, что твои коллеги получат открытый доступ через сеть к твоим личным файлам, или о том, что твоя девушка просмотрит каталог «Мои рисунки» на твоем ноутбуке.



MemOptimizer v3.01

MemOptimizer осуществляет мониторинг системы в фоновом режиме и при необходимости освобождает ресурсы. В результате программа работает быстрее и стабильнее обычного.

- УВЕЛИЧИВАЕТ СКОРОСТЬ РАБОТЫ КОМПЬЮТЕРА, УПРАВЛЯЯ ПАМЯТЬЮ;
- АВТОМАТИЧЕСКИ ПЕРЕКРЫВАЕТ ОСТАТКИ ПАМЯТИ ЗАКРЫВШЕЙСЯ ПРОГРАММЫ;
- ОПТИМИЗИРУЕТ ПАМЯТЬ В КРИТИЧЕСКИХ СИТУАЦИЯХ И СНИМАЕТ РИСК СБОЕВ;
- ВСЕГДА ВИДИТ, КАКОЕ КОЛИЧЕСТВО СВОБОДНОЙ ПАМЯТИ ЕСТЬ В РАСПОРЯЖЕНИИИ.

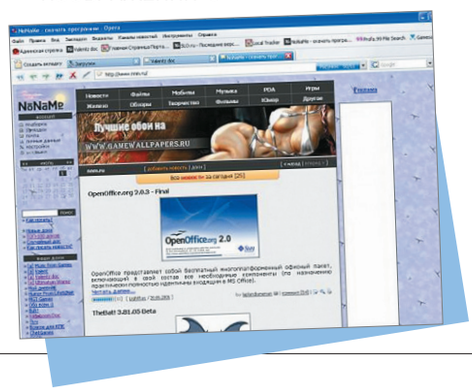


Opera 9.01 Test (8509)

Новая версия популярного в массах интернет-браузера, ставшего теперь бесплатным, Opera; как уверяют разработчики, имеет новый движок и уникальную функциональность; удобный пользовательский интерфейс, стабильность.

В девятой версии браузера представлено огромное число нововведений, к которым добавлены:

- OPERA WIDGETS;
- ПОДДЕРЖКА BITTORRENT;
- ОПТИМИЗИРОВАННЫЙ ПОИСК;
- УЛУЧШЕННАЯ БЛОКИРОВКА ВСПЛЫВАЮЩИХ ОКОН И ИНФОРМАЦИИ НЕЖЕЛАТЕЛЬНОГО СОДЕРЖАНИЯ;
- ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР ИЗОБРАЖЕНИЙ С



Soft

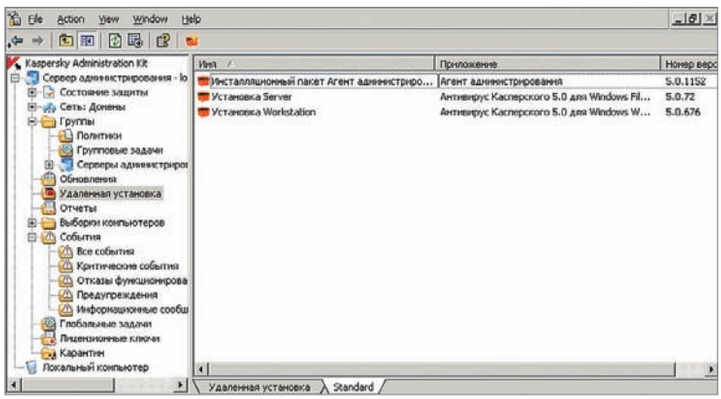
admining

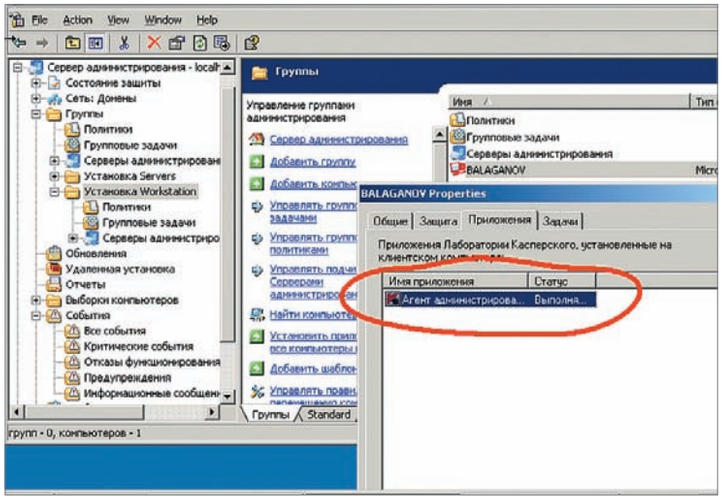
НАСТРОЙКА АНТИВИРУСА КАСПЕРСКОГО. СОЗДАНИЕ ГРУПП, ЗАДАЧ И ПОЛИТИК

АЛЕКСАНДР ПРИХОДЬКО
(SANPRIH@MAIL.RU)

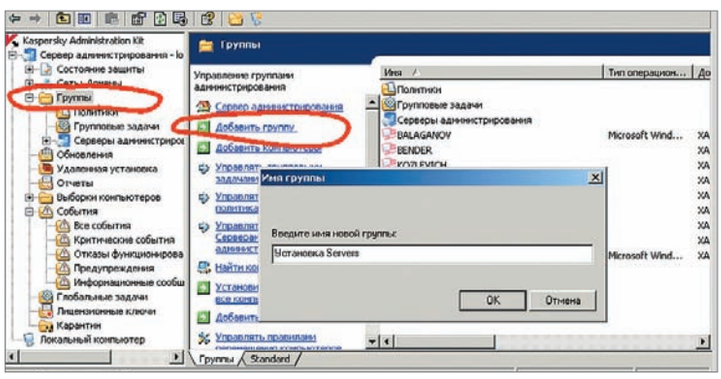
Прежде всего необходимо позаботиться об установочных пакетах. Если в сети присутствуют другие серверы — необходимо создать пакет для установки на операционную систему «Windows Server». При установке как «Kaspersky Administration Kit», так и «Антивируса Касперского для Windows File Servers», программа распаковывала файлы по следующему пути: «C:\kav», там же и лежит первый установочный пакет для операционной системы «Windows Server». Подключим его в «Kaspersky Administration Kit». Для начала запустим «Kaspersky Administration Kit»: «Start» → «Programs» → «Kaspersky Administration Kit» → «Kaspersky Administration Kit». Кстати, ярлык лучше вытащить куда-нибудь поближе: либо на панель запуска, либо на «Рабочий стол». В «Kaspersky Administration Kit» выбираем папочку «Удаленная установка». Щелкаем по ней правой кнопкой мыши — «New» → «Инсталляционный пакет». Запускается мастер создания установочного пакета. Даем имя инсталляционному пакету: так как мы создаем инсталляционный пакет для операционной системы сервера, то назовем пакет «Установка Server». «Next». Следующее окно предлагает выбрать дистрибутив приложения для установки. Нажимаем кнопочку «Обзор» и лезем на диск «C:\kav\WinFileServers\russian» (именно туда распаковывался по умолчанию установочный пакет «Антивируса» для операционки «Server»). Установочный пакет называется «fileserver.kpd». Выбираем его, нажимаем кнопку «Open».

Нажимаем «Next». Следующий экран мастера предлагает нам ввести путь к файлу лицензии. Аналогичным образом указываем путь к файлу лицензии на право обладания продуктом «Антивирус Касперского для Windows File Servers». После нажатия кнопки «Next» начинается загрузка. Кстати, инсталляционный пакет для сетевого агента уже присутствует в админките. Теперь нам для счастья необходимо создать пакет для «Антивируса Касперского для Windows Workstation». Берем дистрибутив антивируса для рабочих станций (на момент написания статьи самый свежий дистрибутив назывался «kav5.0.676_winwk-sru.exe») и запускаем его на установку на сервере. Нас интересует только распакованная программа установки. Распаковывается она туда же: «C:\kav\» в папку «WinWorkstation». Получаем предупреждение о невозможности установки «Антивирус Касперского для WindowsWorkstation» на компьютер под управлением серверной версии «Microsoft





- Инсталляционные пакеты
- Выполнение агента
- Создание групп



Windows». Да не особо-то и хотелось! Мы получили в руки распакованный дистрибутив для «Windows-Workstation». Теперь подключим установочный пакет в админките. Проделываем такую же операцию, как и с пакетом для «Сервера»: «Kaspersky Administration Kit» → выбираем папочку «Удаленная установка». Щелкаем по ней правой кнопкой мыши — «New» → «Инсталляционный пакет». Называем пакет «Установка Workstation», указываем путь к пакету, к ключу — и пакет установлен и готов к эксплуатации.

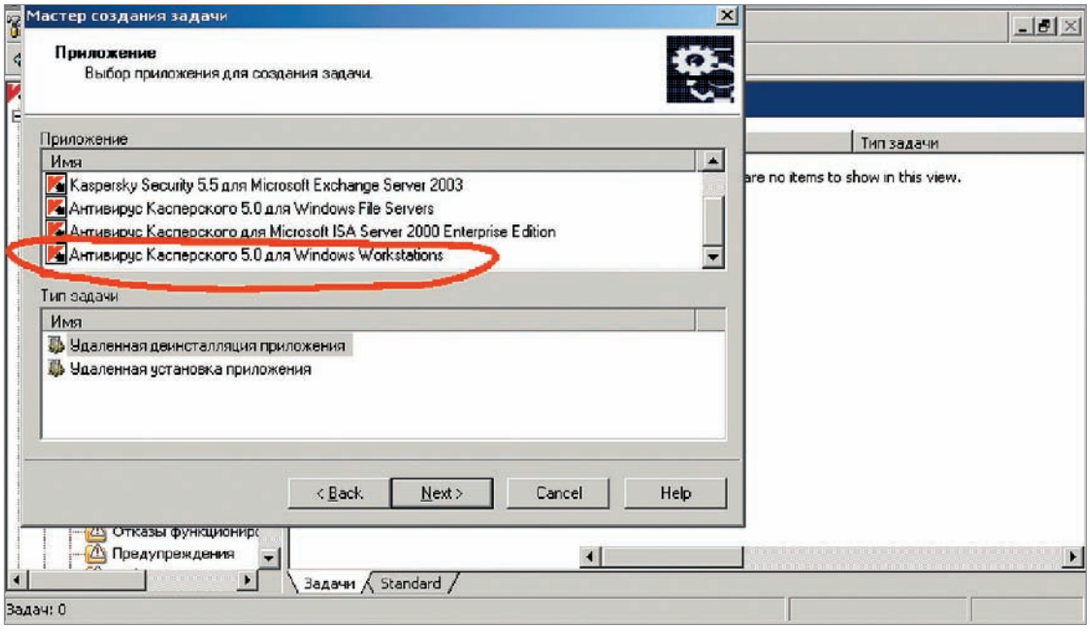
Все инсталляционные пакеты, а также пакеты будущих обновлений лежат здесь: «C:\Program Files\Kaspersky Lab\Kaspersky Administration Kit\Share\Packages». При установке «Kaspersky Administration Kit» программа установки автоматически расшаривает папку «Share» и дает к ней полный доступ группе «Everyone». Проследи, чтобы так все и было. Мы почти готовы к построению сети антивирусной защиты. Для счастья необходимо еще немного поработать над настройкой админкита.

Перед тобой стоит несколько задач. Первая задача — разворачивание самого продукта на разных операционных системах. Вторая — настройка рабочих систем для операционных систем «Сервер» и «Workstation». Я предлагаю создать некоторое количество групп для разных задач. Идея такова: у тебя в сети появляется компьютер с операционной системой «Server», ты помещаешь его в группу для установки антивируса на операционную систему «Server». После установки ты перемещаешь этот компьютер в группу «Серверы», и он начинает существовать в рамках политики для серверов. Проделаем это. В «Kaspersky Administration Kit», в папке «Группы» выбираем меню «Добавить группу» и даем ей имя «Установка Servers».

Также создаем группу «Установка Workstation». Еще одно лирическое отступление: в каждой группе существуют две папки: «Политики» и «Групповые задачи». Политика определяет набор правил для работы антивируса, и она пока нас не интересует. Создадим задачу для разворачивания антивируса по сети на рабочих станциях. Заходим во вновь созданную группу «Установка Workstation» → «Групповые задачи» → меню «Добавить задачу». Радостно хлопаем в ладоши, так как опять появился мастер. Далее все по сценарию: даем имя задаче, а в следующем окне нас поджидает сюрприз в виде приложений, из которых и можно создавать задачи.

Выбираем «Kaspersky Administration Kit» → «Тип задачи» → «Удаленная установка приложения» → «Next» → «Установка Workstation» (если помнишь, этот инсталляционный пакет создали мы сами). Затем мастер предлагает выбрать учетную запись. Вот здесь остановимся на минутку. Предполагается разворачивать антивирус по сети без нашего вмешательства, следовательно, учетная запись должна иметь полные права на все компьютеры сети, то есть должна использоваться учетная запись локального Администратора. Обычно сисадмины, чтобы избежать распухания своей головы от количества паролей, оставляют на всех рабочих станциях локальную учетную запись «Администратор» и придумывают для этой записи один унифицированный пароль. Вот теперь пришло время воспользоваться именно этой учетной записью локального Администратора.

В окне с расписанием запуска задачи оставляем способ запуска «Вручную». Ты ведь не хочешь, чтобы процесс протекал без твоего контроля? Таким же образом создаем еще одну задачу: установка сетевого агента. «Установка Workstation» → «Групповые задачи» → меню «Добавить задачу» → приложение «Kaspersky Administration Kit» → «Тип задачи» → «Удаленная установка приложения» → «Next» → «Инсталляционный пакет Агент администрирования версии 5.0.1152» → опять выбираем учетную запись локального Администратора, вводим пароль, в расписании выбираем «Вручную» и завершаем работу мастера. Мы готовы к разворачиванию антивируса по сети для «Windows Workstation». Так развернем же его! Добавляем машину Балаганова в группу «Установка Workstation». Наступаем на папку «Группы», берем левой кнопкой мыши машину «Balaganov» и перетаскиваем ее в группу «Установка Workstation».



Выбор приложения

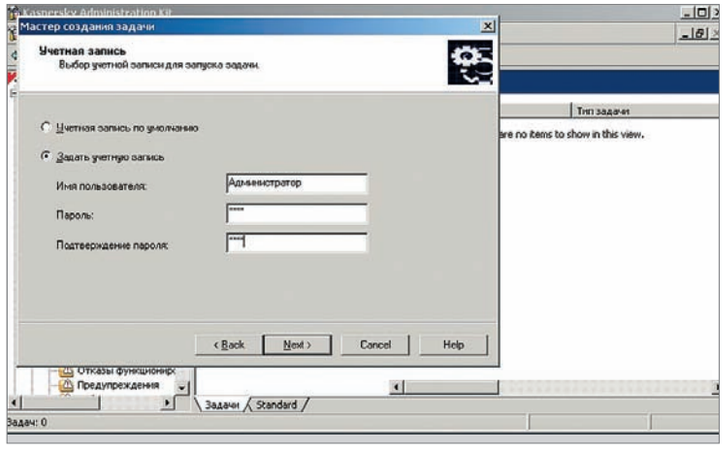
Перенос компьютера в группу установки

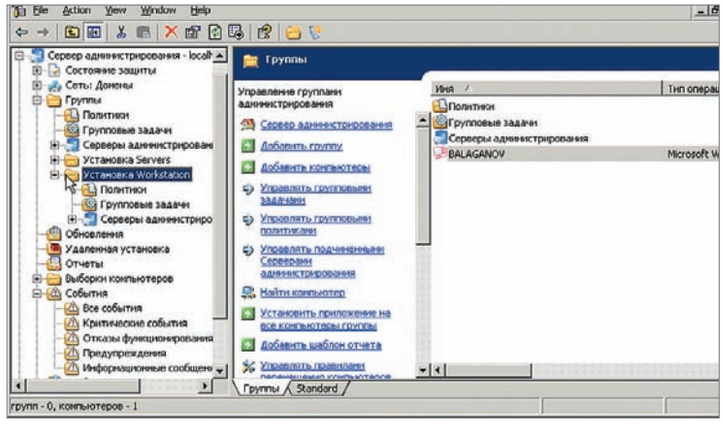
Сначала установим на компьютер сетевого агента, а затем и сам антивирус. Правая кнопка мыши на задаче «Установка агента» → «Запустить». Через какое-то время на панели задач компьютера Балаганова появляется задача установки сетевого агента.

Теперь посмотрим, что админкит знает про машину Балаганова. Двойной щелчок мыши на машине Балаганова в админките вызывает свойства этой машины: переходим на закладку «Приложения» и видим, что сетевой агент администрирования успешно установлен и работает.

Пока все идет по плану. Установим теперь на комп Балаганова антивирус. Правая кнопка мыши на задаче «Установка Workstation» → «Запустить». Ждем результатов. Контролировать процесс установки можно через свойства задачи → кнопка

Выбор учетной записи



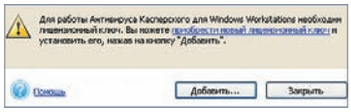


«Результаты» → двойной щелчок в левом поле на компьютере Балаганова, и читаем информационное сообщение о необходимости перезагрузки.

После перезагрузки на машине Балаганова уже будет стоять антивирус Касперского. Все это я так подробно описал только для того, чтобы ты понял, что в принципе на этом вся работа и заканчивается! Проведешь механизм установки на одной машине, а затем все рабочие станции перетащишь оптом в группу «Установка Workstation», а все серверы — в группу «Установка Server» и запустишь созданные тобой задачи. Однако расслабляться еще рано. После перезагрузки компьютера Балаганова мы получаем неприятный сюрприз. И это вполне закономерно — мы же не «прикрутили» лицензионный ключ!

Возвращаемся в папку «Групповые задачи» и создаем задачу установки лицензионного ключа. Меню «Добавить задачу» → даем задаче имя «Установка ключа» → «Next» → теперь из приложений выбираем «Антивирус Касперского 5.0 для Windows Workstation». «Тип задачи» → «Установка лицензионного ключа» → «Next» → натравливаем мастер на ключевой файл, отмечаем галочкой «Использовать в качестве текущего лицензионного ключа» → выбираем учетную запись локального админа → расписание → «Вручную» — и задача создана. Запускаем задачу установки ключа. И вуаля — у Балаганова запустился антивирус Касперского. Точно такие же задачи создаем в группе «Установка Servers», естественно, с правильными пакетами и ключами. Одна маленькая поправка: мы только установили антивирус, но не производили его настройку. Для настройки антивируса под наши нужды мы создаем еще две группы: «Серверы» и «Рабочие станции». Процесс создания групп я опускаю.

В новых группах нас больше интересуют политики. В группе «Рабочие станции» создадим политику,



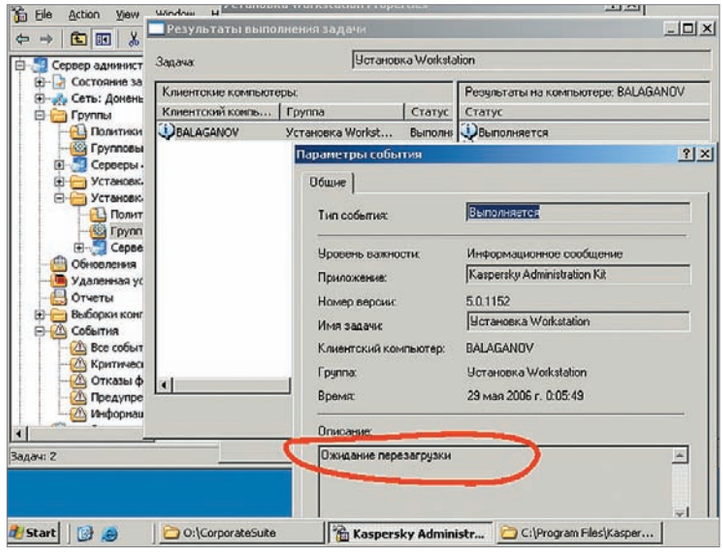
Отсутствие лицензионного ключа

Информационное сообщение «Ожидание перезагрузки»

регламентирующую работу антивируса. Заходим в группу «Рабочие станции» → папка «Политики» → меню «Добавить политику» → опять мастер → «Next» → имя политики «Настройка защиты» → «Next» → в окне приложений выбираем «Антивирус Касперского 5.0 для Windows Workstation» → «Next» — заслуженный отдых. На следующем экране выбираем уровень защиты. Но не он сейчас важен (его мы всегда можем поправить). Я хочу объяснить значение замочка.

Замочки запрещают конечному пользователю менять что-либо в настройках антивируса! И это очень важно, так как пользователь не может ничего изменить в пику твоей политике. Нажимаем на замочек — «Next». Далее следует выбрать уровень защиты. Сам уровень пока оставляем по умолчанию, а вот «Действия над обнаруженными объектами» изменим: для пункта «Опасный объект» выбираем «Лечить, а если невозможно — удалять». И опять замыкаем замочки. По умолчанию Лаборатория Касперского уже настроила антивирус, но, если захочешь изменить настройки, дави на кнопку «Настройка» и правь по своему вкусу. Я обычно отмечаю все.

Следующее окно — настройка обновлений. Выбираем источником обновлений только «Сервер администрирования». При такой настройке машина, на которой развернут «Сервер администрирования», будет



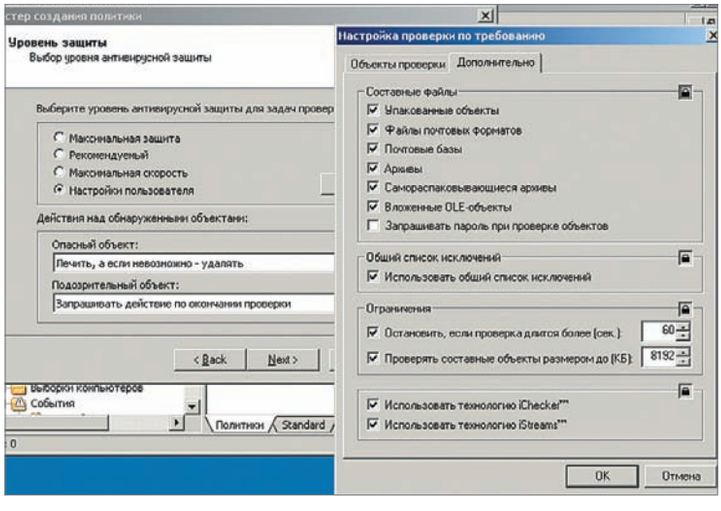
закачивать обновления и раздавать их всем клиентам. В противном случае у тебя все рабочие станции будут лезть в интернет для получения обновлений. Зачем расходовать трафик? Если оставить отмеченным пункт «Серверы обновлений Лаборатории Касперского», то клиентские машины будут по своему расписанию синхронизировать обновления и с «Сервером администрирования» ломиться в интернет. В этом же окне нажимаешь кнопку «Параметры LAN» и прописываешь свой прокси-сервер. В окне «Параметры обновлений» убираешь галочку с «Запрашивать подтверждение перед установкой» (все равно твой юзер не поймет, о чем идет речь, а тебе лишние вопросы ни к чему), замыкаешь замочек — и политика создана! Теперь перетаскиваешь машину Балаганова из группы «Установка Workstation» в группу «Рабочие станции», и его антивирус ляжет под настроенную тобой политику. Теперь, если ты дважды щелкнешь по вновь созданной политике, ты увидишь еще целое море свойств, по которым тебе необходимо пройтись и позакрывать замочки. Кстати, ты можешь убрать интерфейс антивируса с машины пользователя, и он понятия не будет иметь о том, что у него установлен антивирус.

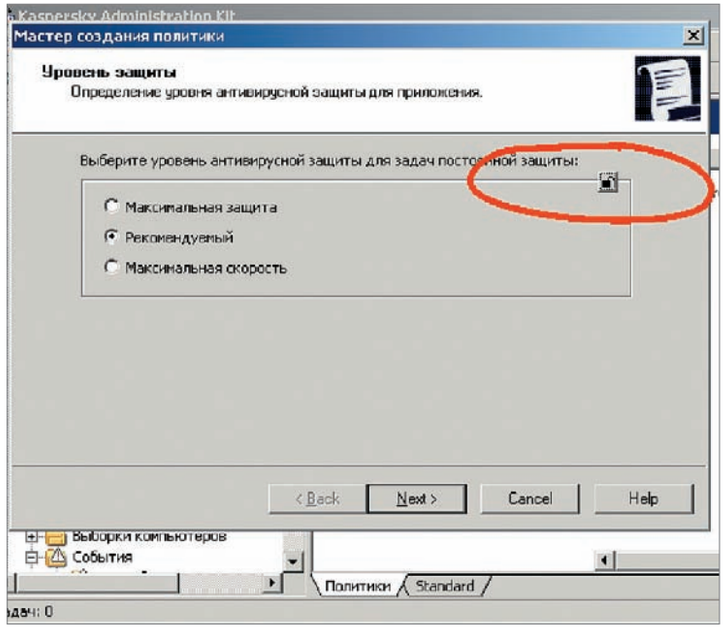
Что хотелось бы сказать еще: в админките существует иерархия применения политик. Так, например, если создать какую-либо политику на корневой группе «Группы», то эта политика будет действовать абсолютно на все вложенные группы. Это было бы хорошо, если бы у нас была однородная сеть и не существовало бы никаких нестандартных задач. Однако на практике возникают разные потребности, поэтому, создавая группы по операционным системам и другим признакам, ты оставляешь себе место для маневра.

В следующий раз мы продолжим ковырять настройки антивируса: настроим политику для серверов, задачи обновления, посмотрим на альтернативные способы установки приложений, познакомимся с отчетами ●

Запрет изменений

Настройки уровня защиты





НЕ ХВАТАЕТ ЧЕГО-ТО ОСОБЕННОГО?

Играй просто!
GamePost

Final Fantasy XI: The Vana'diel Collection (US Version)
$69.99

Call of Duty 2 Collector's Edition
$99.99

Command & Conquer: Collection
$49.99

* Diablo Action Figure: **Necromancer** **$42.99**

У НАС ПОЛНО ЭКСКЛЮЗИВА

- * Эксклюзивные игры
- * Коллекции фигурок из игр
- * Коллекционные наборы

Тел.: (495) 780-8825
Факс.: (495) 780-8824
www.gamepost.ru

GAMEPOST

Crew

e-mail

ПИШИТЕ ПИСЬМА! SPEC@REAL.XAKEP.RU
SKYWRITER

viftik@mail.ru
на майле
help ми кто-нибудь

Здравствуйте!
У Вас на DVD (за номером #076) был выложен видео ролик по «Hydra», это там где амерекосовское мыло сбручивали...

Так вот, не могли бы Вы сказать, чья музыка использовалась в качестве сопровождения в клипе, дело в том, что диск похерил один «редиска»...

244931736 — если удобней ася к Вашим услугам. Заранее благодарен.
С уважением.

Уважаемый Вифтик!
Я долго крутил-вертел диск Спец'а в руках, искал на нем 4,7 Гб данных и прочие вкусности. И так посмотрел, и этак — никаких признаков DVD, обычный CD. Параллельно в моей голове родился второй вопрос: мы, вроде бы, никаких видеороликов, кроме порноредсостава, в архиве-сюрпризе никогда не выкладывали. В общем, не побоюсь показаться чересчур умным и предположить, что ты, видимо, не на тот адрес послал мыло.
P.S. Асю брутим. Заранее благодарны.

vovka_@gmail.ru
объявление

Уроды, _распутная женщина_!
С интересом прочитал в «спеце» N65 статью «Секретная лаборатория», в которой сотрудник «Лаборатории Касперского» _очлененно_ _расженскополовоорганнивает_, какой _женскополовоорганный_ случай помог быть этому, типа, проекту.

Так вот, _член_, это была жена (бывшая) товарища Касперского и, если бы не она, этот _любитель секса_ так бы и остался нищим, каких _до члена_!

Конечно, потом он эту жену кинул натурально (еще бы, [тут автор выражает сомнения в человеческих качествах героя]!) и взял себе естественно помоложе, там и _бюстгалтер побольше_!

А вообще, товарищ Касперский (вернее команда _любителей секса_ под его предводительством) пишут исключительно [тут автор выражает сомнение в качестве ПО] и бажные проги, которые тормозят всю систему!

Еще раз повторю, если бы не его жена, так он бы и остался нищим человеком, каких _до члена_!
P.S. пишу так потому, что вы все равно «облажаете» и стиль письма, и меня.

Владимир, здравствуй!
Очень хотелось опубликовать твое письмо без цензур, купюр и исправлений — уж очень оно изобилует экспрессией и красотой. Но, к сожалению, нецензурная лексика в нашем издании недопустима, поэтому пришлось слегка (ммм, насколько это вообще было возможно сделать «слегка») отредактировать твое творчество, надеюсь, ты не обидишься и не подумаешь, что тем самым мы хотели тебя «облажать».

Несомненно, высказанные замечания имеют огромную пользу и влияние как на качество ПО лаборатории Касперского как таковое, так и на их продажи. В связи с этим, на случай, если господин Касперский не читает нашего издания, твое творчество мы переслали ему лично.

Что касается личной жизни Касперского, то тут для меня и вовсе тайна, покрытая мраком. Ты же, как я вижу, уже можешь снять небольшой, но очень трогательный сериал на эту тему. Что тебе и рекомендую сделать в ближайшее время, так, чтобы мы смогли выложить его на диск. Уверен, что это понравится пусть и немногочисленной, но все же присутствующей части наших читателей — девушкам.
Спасибо, Владимир!
Рады были, что ты про нас не забываешь.

mongol2014@mail.ru
ЯРИК
не указана

Я хочу установить Windows версией ниже, например, Windows 2000. Я все пробовал, но ничего не получается, я удалял полностью Windows MCE, в биосе ставлю загружаться с диска, в сидиром вставляю диск, но комп его не видит, а когда вставляю диск с Windows MCE, то он начинает устанавливать его (Windows MCE). Пожалуйста, помогите мне, на вас последняя надежда, очень прошу....

Хм... Знаешь, анекдот есть такой:
— Доктор, когда я вот так делаю, у меня болит.
На что доктор меланхолично отвечает:
— А Вы так не делайте...

Зачем тебе Windows версией ниже? Или ностальгия замучила? Тогда советую Windows 3.1 — полное ощущение погружения в Матрицу. Ну, да ладно, не наше, редакционное это, в общем-то, дело. Вопрос о том, что при установке диска Windows MCE устанавливается Windows MCE я вообще отбрасываю как несостоятельный — глупо было бы ожидать чего-то иного, верно? А что до диска Вин2000, могу тебе порекомендовать проверить одну простую вещь — проверь, загрузочный ли он вообще. И, если все-таки загрузочный, но Винда с него не грузится, советую раздобыть дисковод и сделать загрузочные дискетки для установки (тебе понадобится 3 штучки).

Как их делать, я, честно говоря, не помню. Но все вертится вокруг ключиков к программе установки (winnt.exe). Имхо. В общем, пилите, копайте и не забывайте сметать стружку, Шура.

ssheglov@mail.ru

Andrey Sheglov
аргхххххххххх

Здравствуйте, скажите, пожалуйста, возможно ли приобрести в редакции старые номера журналов Хакер и Спец Хакер. Если нет, то подскажите, как это можно сделать.
Андрей.

Привет, Андрей!
Черт, иногда мне кажется, что то ли эту рубрику никто не читает, то ли не запоминают ответы на главные вопросы. Просто есть ряд вопросов, на которые мы отвечаем с завидным постоянством как просто письмами, так и непосредственно в номере.

Так вот, милый мой Андрюшка. Ты сорвал Джекпот — задал один из самых популярных вопросов.

В общем, в который раз повторяю: номера журналов за прошлый, позапрошлый и т.д. вплоть до времен, когда динозавры ходили по нашей матушке-планете, можно найти только у букиниста и в библиотеке. Еще вариант — цифровая форма на нашем сайте или файлы на наших дисках. Все. Больше искать тебе их негде, Андрюша.
Удачи в поисках!

urmat2004@list.ru

Урмат Аймамбетов
Пароль

Не могли бы вы прислать пароль на сюрпирз в компашке в Спеце за октябрь 2005 года. Пишу из Кыргызстана, так что я не мог прислать sms.
Заранее спасибо.

Привет, Урмат! Как-то ты... В общем, не хочу тебя обижать, но как-то ты медленно реагируешь. Смотрю вот за окошко и вижу солнышко, лето, в общем, июль 2006 года за окном, понимаешь?
А октябрь 2005 года — это холодно, дождик и листики с деревьев облетают. И, как понимаешь, это давно было, так что, к сожалению, пароля я не помню. Жаль, конечно, но единственный способ теперь получить пароль — это брутить его. Могу посоветовать RAR Password Cracker.
Удачи!

karen@xaker.ru

KAREN
Спасибо

Уже на протяжении двух лет читаю выпуски ваших журналов, можно сказать постоянный читатель. Хочется, во-первых, поблагодарить весь состав Хакера и особенно Хакер Спец, несмотря на критику (иногда объективную) от других читателей по поводу содержания статей журнала, мне нравится содержание обсуждаемых в журнале тем, а особенно содержание прилагаемых дисков. Респект всем, особенно коллегам (программистам).

P.S. Если не трудно в следующем номере или с помощью мэйла проинформируйте, где можно увидеть статьи где проводится сравнительный анализ ОС Винды и Линукс, а там я уже решу ставить все таки тукс или неа.
С уважением Карен — Cyberhacker!

Карен? Казарьян? Ты ли это?
Ну, надеюсь, нет, а то бы я начал подозревать тебя в раздвоении (растроении?) личности.

Большое тебе человеческое спасибо за похвалу, особенно диска — я очень растрогался! Коллеги тоже дружно пожали тебе руку. Скажу тебе — «благодарю»!

P.S. Уважаемый Карен! Сравнительный анализ ОС Винды и Линукс, в общем-то, невозможен. Не буду вдаваться в дебри holywar... Если ты хочешь узнать, чем тебе пользоваться, то очень советую поставить приложение VMWare Workstation и инсталлировать на нее сначала Линукс, посмотреть, попользоваться. Потом попользоваться Виндой. И решить. Никакой сравнительный анализ тебе не поможет.
Ну, удач тебе, Киберхакер.

xxx-satana@inbox.ru

LEXA MEX
Serial number

Здравствуйте, редакция СпецХакер!!!

У меня к вам есть одна большая просьба или точнее вопрос: где можно достать серийный номер на программу Sony ACID Pro 6.0 Build 214? Т.к. на вашем диске не было даже пробной. Или, как там, ознакомительной версии. Помогите, пожалуйста, найти. Заранее спасибо!

Алексей! Видишь ли, на диске ее могло не оказаться по ряду причин: от банальной нехватки пространства до проблем с лицензией на распространение — мы же честные хакеры. Поэтому и методику поиска ключика подсказать мы тебе не можем, хотя все прекрасно знаем (тут ситуация похожа на ситуацию с собакой — она, вроде бы, знает ядерную физику, а в дискуссию вступить не может, ибо языковой барьер!) В общем, рекомендую популярные поисковые ресурсы (**yandex.ru**, **aport.ru** и т.д. — пусть никто не обижается, кого не упомянули. Ключевое слово — «лекарство».
И будет тебе счастье **C**

Story

могиканин

КУЛЯБИН, ЕСЛИ ХОРОШЕНЬКО ПОДНАПРЯЧЬСЯ, МОГ ВСПОМНИТЬ ТОТ ДЕНЬ ЕДВА ЛИ НЕ ДО МЕЛОЧЕЙ. ВПРОЧЕМ, МОГ ВСПОМНИТЬ, ДАЖЕ И НЕ НАПРЯГАЯСЬ...

NIRO (NIRO@REAL.XAKEP.RU)

Тогда была зима, которая и послужила причиной всему. Холодная, отвратительная зима. Кулябин никогда не любил холод, не мог к нему привыкнуть и всегда с нетерпением ждал весны — а тут, как на грех, зима выдалась особенно холодной, ветра — уж слишком пронизывающими, морозы — чертовски крепкими!

Каждое утро Кулябин, проклиная все на свете, топал на стоянку по темным улицам, чтобы забрать свою «Тойоту» и отвезти ребенка в школу. Снег громко скрипел под ногами, заставляя ежиться от этого противного звука; ветер задувал во все приличные и неприличные места, руки мерзли даже в карманах. Стояла самая что ни на есть нелюбимая Кулябиным погода — и он был этой погодой раздавлен, угнетен и практически превращен в сосульку.

Лишь только после того, как двигатель прогревался, и в салон начинал поступать уже достаточно теплый воздух, Кулябин мог позволить себе перестать прятать голову в шарф и немного расслабить напряженные мышцы спины. Заднее стекло оттаивало, изо рта прекращал вылетать пар, из колонок в дверях начинал доноситься нормальный звук, а музыкальный центр, вместо того чтобы выплевывать нераспознанный запотевший диск, начинал, наконец, нормально читать файлы, и можно было выключить глупое местное радио и включить что-нибудь удобоваримое — Поля Мориа, Джо Дассена или Криса де Бурга.

Наиболее ответственным во всем этом алгоритме был, безусловно, пуск двигателя. Кулябин выучил эту процедуру за четыре зимы. Подойти, осмотреть машину на предмет ударов ее чужими дверями (с некоторых пор в соседях по стоянке он перестал замечать аккуратность), потом открыть дверь и, если накануне выпал снег, с силой закрыть ее снова — чтобы осыпался снег, который мог бы при неудачном раскладе попасть в салон. Потом дверь открывалась опять, включались фары — ненадолго, секунд на тридцать. Знающие люди уверяли, что это усиливает реакцию в электролите аккумулятора — и хотя Кулябин был человеком с высшим образованием, и кое-какие познания в химии и физике у него остались, — эта часть процедуры казалась ему чем-то сродни мифу: кто ее измерял, реакцию-то? Но пропустить этот пункт он уже, в силу традиций, не мог. Щелчок — и фары уже выключены. Теперь можно и ключ повернуть — ненадолго, на несколько оборотов двигателя, чтобы топливо в цилиндры попало, но свечи не смогли его воспламенить. Затем секундная пауза, после чего — снова поворот ключа — на этот раз до тех пор, пока не заведется. «Дыр-дыр» — и через пять, максимум десять секунд все в порядке. Машина за много лет не подвела его ни разу — но все равно, при повороте ключа на «Пуск», он замирал в ожидании того, что двигатель не заведется.

И ИЗ НЕГО ТОНЕНЬКИМ РУЧЕЙКОМ ПОЛИЛАСЬ КИСЛОТА, КАПЛЯ ЗА КАПЛЕЙ

Так они и играли с машиной — кто кого. Но однажды пришли жуткие морозы... Машин по утрам на стоянке оставалось с каждым утром все больше и больше — у кого-то оказалось не то масло, у кого-то сдохли свечи, у кого-то — аккумулятор. В то злополучное утро проиграл свою битву за зажигание и Кулябин — впервые за несколько удачливых зим.

Мотор не хотел заводиться. Не хотел — и все. Кулябин побоялся посадить аккумулятор полностью, бросил попытки справиться с непокорной «Вистой», позвонил домой — и жена отвезла сына в школу на такси.

Кончилось тогда все хорошо — после работы он пришел на стоянку, вместе с охранниками они руками затолкали машину на мойку, которая была пристроена к охраняемой территории. И на мойке знающие мальчишки отогрели двигатель струей горячей воды, направленной куда-то в недра блока цилиндров. Уже через пятнадцать минут мотор завелся так, словно на улице было лето — но седьмое чувство подсказывало ему, что расслабляться рано — и он поехал на станцию техобслуживания, сменил для верности масло, купил свечи с платиновыми наконечниками и приобрел новый аккумулятор.

Вот тогда-то и случился первый толчок — что-то, похожее на жадность, но носящее маску практичности, заставило засунуть его старый аккумулятор в багажник. Он не мог объяснить, зачем сделал это, но жена его поступок одобрила, тоже особо не вдаваясь в объяснения. Всю зиму и начало весны Кулябин прокатал в машине старый свинцовый параллелепипед, вспоминая про него только тогда, когда открывал багажник, а случалось это крайне редко. Возить ему там было нечего, страстью к перевозке картошки в мешках или рассады он не отличался, на природу выезжать не любил — поэтому там всегда было пусто и чисто.

А потом наступило лето. В июне Кулябин получил приличный аванс за свой очередной проект — и жена тут же решила вложить упавшие с неба деньги в новое авто. Кулябину это показалось совершенно ненужным — старая машина вполне его устраивала, но вторая половина так не считала. Несколько дней споров на эту тему он еще выдержал, но когда началась следующая неделя — сдался. Так долго спорить об одном и том же он не мог — пришлось уступить.

Все случилось тогда, когда они стали готовить машину к продаже. Ничто не предвещало никаких проблем — двигатель работал как часы, ходовая часть не подвела ни разу, в салоне — чистота. А вот в багажнике...

Жена открыла багажник... После чего остановить ее красноречие Кулябин уже не смог.

Аккумулятор, конечно, был пристроен в багажнике очень прочно — исходя из центра тяжести машины и возможных перегрузок при движении. Но, к сожалению, и на старуху бывает проруха. В какой-то из моментов езды по «русским горкам» — а именно на это похожа основная часть наших дорог — аккумулятор упал набок. И из него тоненьким ручейком полилась кислота... Капля за каплей...

От покрытия в багажнике осталось его жалкое подобие. Складывалось ощущение, что там еще с зимы поселилась гигантская моль, которая грызла все, что попадается на пути. И эта самая моль сожрала и коврик, и квадратный метр краски в нише рядом с запасным колесом, и еще много чего — того, что, по большому счету, не стоило больших денег... Но не все можно измерить деньгами.

И жена уже не помнила, что тоже приложила руку к тому, чтобы аккумулятор оказался в машине и был забыт там на целых полгода. Она не помнила, что половина денег, вложенных в эту машину, была заработана ее мужем. Она просто кричала на него, как одержимая, проклиная тот день и час, когда познакомилась с Кулябиным и решила связать с ним свою судьбу. Кричала пять минут, десять, пятнадцать... Кулябин слушал все это и понимал, что цена ремонта машины несопоставима с ценой его собственной жизни.

И когда она внезапно остановилась, чтобы отдышаться от своего монолога, он повернулся к ней спиной и ушел. Совсем.

Вот ведь какая цепочка получилась... Зимний день... Мороз... Заглохший мотор... А обернулось все летним скандалом, определившим его дальнейшую судьбу. Его и еще семерых человек, о существовании которых он не имел ни малейшего представления.

В тот день, когда он остался один, он совершил ошибку. Через три часа после скандала, послужившего причиной развода. Ошибку, которую не смог сразу заметить и исправить.

А все потому, что в феврале были слишком уж крепкие морозы.

* * * * *

Он стоял в очереди за пивом — ежедневный ритуал. Правда, приходилось следить за собой, чтобы не опуститься окончательно, ибо работа требовала всех его нервных клеток; выпить ровно столько, чтобы снова стало легко, чтобы забыть грусть, которая копилась в нем годами... Он знал, что забудет ненадолго — ровно настолько, чтобы заснуть с чистой совестью; а завтра новый день, работа, потом снова очередь в ларек, бессмысленные разговоры с такими же, как он, людьми, у каждого из которых свой скелет в шкафу. Рыба, соленые орешки, кальмар, еще пиво... Ежевечерний круговорот. Что ему нравилось в пиве — постепенное, медленное опьянение; глоток за глотком он прогонял мрачные воспоминания, оставляя на дне пластикового стакана свои слезы и горечь одиночества. Так было, так есть и так будет. Сегодня, завтра и всегда...

— ...Кулябин Дмитрий Анатольевич — это вы?

Рядом стоял молодой, но уж с очень взрослым, не по годам, взглядом, человек в камуфляже без знаков отличия в погонах и петлицах. Вполне возможно, вообще не военный, — но нет, было в нем что-то такое, что не давало ни секунды сомневаться в его принадлежности к Министерству обороны.

Кулябин кивнул. Хотелось отхлебнуть пива из только что купленного стакана; пена еще не осела, он держал его на отлете, чтобы случайный порыв ветра не сдул белые пенные кружева на костюм. Человек, назвавший Кулябина по имени, молча смотрел на него, словно забыв о цели разговора. Пальцы на руках сжимались и разжимались, похрустывая, но Кулябин чувствовал, что военный хотел не ударить, а просто не знал, как продолжить беседу.

— Вы меня не знаете, — внезапно сказал незнакомец.

— Точно, — кивнул Кулябин и, наконец-то, решился отхлебнуть пива. — Может, представитесь? И тогда будет повод выпить за знакомство.

— Мое имя Андрей... И я говорю его Вам просто потому, что надо как-то обращаться ко мне. Лучше по имени. Остальные данные — ни к чему.

— Тайны, тайны, — пробурчал в стакан Кулябин. — Что вам нужно от меня, Андрей? Судя по всему, вы человек военный...

— Так точно, — кивнул он в ответ. — Военный. Можно даже сказать, секретный. То есть засекреченный. Был.

— Был? И что же случилось? Давайте отойдем к столику, место освободилось, — предложил Кулябин. Андрей кивнул — не очень удобно было разговаривать посреди очереди. Они подошли к высокому столику, заставленному одноразовыми грязными тарелками с остатками рыбы и кучей ореховой шелухи; Андрей брезгливо посмотрел на все это, Кулябин же, совершенно не задумываясь, смахнул тарелки в пластиковый мешок, подвешенный к столику снизу.

— Да ничего, собственно говоря, не случилось, — сказал Андрей, глядя на Кулябина, сосредоточенно потягивающего пиво. — Так, мелочь... Вы слышали про недавний инцидент, происшедший на наших северных границах? Думаю, что слышали — про него много рассказывали в новостях, мусолили подробности в прессе...

Кулябин на пару секунд наморщил лоб, напрягся, но вспомнить ничего не смог.

— Я газет давно не читаю, телевизор смотрю редко. Знаете, все больше книги...

— Ну да, конечно, — согласно кивнул Андрей. — У каждого свой способ убить время. Вы не обязаны были знать об этом происшествии. Следуя утверждению о том, что все люди разные, мне надо было сразу представить себе, что такой человек, как вы, вряд ли следит за новостями. Скажите, а есть в вашей жизни хоть что-нибудь...

— Есть, — внезапно произнес Кулябин, поставив стакан с пивом на стол. — Есть. Вы чего ко мне подошли? В душу залезть? Какого черта? Что вам от меня надо? У вас у самого — какие ценности в жизни?!

— Уже никаких, — отведя глаза в сторону, ответил Андрей спустя некоторое время. — Были... Был... Друг. Теперь его нет. А семьей обзавестись не успел. Родители уже умерли, отец — еще когда я пацаном был, мать, вот, недавно... На прошлой неделе была годовщина. Третья уже по счету. Сестра вот еще... Да, у нее свои проблемы.

— Простите, — сказал Кулябин, — я, конечно же, не мог знать ничего этого. Я и сам, в общем... Давно без семьи. То есть... Родители далеко, видимся редко в силу дороговизны перемещений по России. Так, перезваниваемся иногда. А жена ушла. Сына забрала. И никаких координат. Уже тоже вот — три года. Три года.

Он замолчал, отхлебнул пива. Андрей смотрел на него молча, словно оценивая все те слова, что услышал сейчас. Потом спросил:

— Три года? Странное совпадение, не правда ли?

— Да уж... — натужно улыбнулся Кулябин. — С детства не верю в совпадения.

— Дмитрий Анатольевич, совпадения — это неосознанная закономерность. У вас жена три года назад ушла, у меня... Скажите, «Могиканин» — это ваш проект?

— Да, — не задумываясь, ответил Кулябин. — Мой. Но откуда вы про него знаете? Вот же пиво, черт возьми, язык развязывается... Секретная информация. Я вам больше ничего не скажу и буду вынужден сообщить о нашей с вами встрече в соответствующие органы безопасности.

— А мне от вас ничего и не надо, — сурово ответил Андрей. — Просто хотелось на вас посмотреть. Типичная ситуация. Прямо как в книгах...

— Посмотрели? А теперь идите, идите побыстрее и сделайте так, чтобы я вас очень скоро забыл — и как вы выглядите, и что ваше имя Андрей, и даже в какую сторону вы отсюда пойдете.

— Пойду. Чуть позже. Я хочу сказать вам, что на вашей совести жизнь моего друга. И еще пятерых человек, которые не были моими друзьями, но оставили след в моей жизни. Неизгладимый след. Вот вы здесь, живой, пьете пиво и радуетесь жизни, а они...

— Я? Радуюсь жизни? — искренне удивился Кулябин. — Я похож на человека, который радуется жизни? Судя по всему, вы совсем не знаете сущности того предмета, который называете жизнью. Похоже, вы пережили какую-то драму, что-то страшное, оставившее внутри вас мину замедленного действия. Я верю вам, — но не понимаю, по какой причине вы взваливаете на меня чужие смерти. И еще — мне кажется, что эта мина с часовым механизмом внутри вас готова взорваться в самые ближайшие

минуты. Не хотелось бы, чтобы меня погребло под обломками вашего разума.

Он посмотрел на опустевший примерно на половину стакан, взглянул на длинную очередь к ларьку, покачал головой.

— Кто-то где-то погиб... Надо обязательно прийти, связать все это с проектом «Могиканин» и пытаться воззвать к моей совести. Надо же, пророк в своем отечестве по имени святой Андрей! Не судите да не судимы будете!

Несколько человек оглянулись на его монолог. Он понял, что почти кричал, и втянул голову в плечи.

— Я радуюсь жизни... Чушь собачья, — прошептал он себе под нос, постукивая ногтем по стакану, потом отхлебнул столько, что даже не поместилось в рот, закашлялся. — Кто Вы, черт Вас дери?!

— Я? Солдат, — ответил Андрей. — И те, кто погибли, тоже были солдатами. Хорошими солдатами.

— Я тоже хороший, — подняв мутный взгляд на Андрея, прокомментировал Кулябин. — Только не солдат. Программист. Понимаете, программист! Я не могу никого убить! И не надо валить все в кучу! Это ваши солдатские проблемы!

— У меня сложилось впечатление, что вы знаете, о чем идет речь, — покачал головой Андрей. — Я здесь не ради решения моральных проблем, не ради привлечения вас к ответственности. Я просто пришел посмотреть на вас. На человека, который... Который сделал то, что сделал. Я, хоть и солдат, но за все время своей службы не убил ни одного человека — и вряд ли смог бы это сделать. Я имею в виду — глядя противнику в глаза...

— Я понял, — внезапно прояснившимся взглядом посмотрел на Андрея Кулябин. — Я понял — насчет глаз... Вы из тех военных, которые не стреляют. То есть стреляют, но — нажимая кнопки. Шлеп по пульту — и из шахты в небо несется нечто огнедышащее. Эдакий дракончик с ядерной начинкой.

— Вы правы, Дмитрий Анатольевич, — согласился Андрей. — Шахта, боевое дежурство, ракеты... Вы ведь понимаете, что если я об этом говорю вполне открыто — значит, никакой военной и государственной тайны в этом нет. И я всегда был готов нажать ту самую кнопку по первому же приказу. И один раз я стоял на краю... Мы все стояли на краю. Из семерых остался в живых я один. И я до сих пор не знаю — убили ли вы шесть человек или спасли весь мир...

— Цена жизни этих шестерых приравнивается к стоимости целого мира? — недоумевающе поднял брови Кулябин. — Молодой человек, сконцентрируйтесь — я пока еще ничего не понял из ваших слов. Кроме, пожалуй, того, что меня убивать вы не собираетесь.

— Не собираюсь, — согласился Андрей. — Не уполномочен. К счастью...

— Да уж... — Кулябин допил пиво. — Я, наверное, буду еще. А вы?

— Нет, Дмитрий Анатольевич, спасибо, — отрицательно покачал головой собеседник. — Я пойду. Хочу, чтобы вы запомнили — ваш проект «Могиканин» был применен по назначению. По боевому назначению.

Кулябин застыл на полпути к ларьку. Потом медленно повернулся к Андрею и переспросил:

— То есть? Вы хотите сказать, его поставили на боевое дежурство?

— Так точно, — глядя ему в глаза, ответил Андрей. — Шестого июля две тысячи седьмого года проект «Могиканин» установлен на компьютерах шахты с порядковым номером двадцать шесть на полуострове Таймыр. Две недели он находился в режиме тестирования, после чего был загружен на главный терминал. Никаких проблем с обслуживанием электронной системы боевого дежурства не возникало — она оказалась продумана до мелочей. Весь персонал шахты — все семь человек — изучили его досконально и в любой момент могли заменить друг друга на боевом посту и нанести по врагу ядерный удар...

СТАКАН ВЫПАЛ ИЗ ЕГО РУКИ НА ЗЕМЛЮ, НО ОН НЕ ЗАМЕТИЛ И ЭТОГО...

Кулябин слушал его, совершенно забыв о том, что хотел купить еще пива. Стакан выпал из его руки на землю, но он не заметил и этого. Человек напротив сообщал ему вещи, о которых он не имел ни малейшего понятия. Хотя...

— И не делайте вид, что вы не знали этого, — сурово сказал Андрей — он будто угадал мысли Кулябина. — Вы, конечно, не относились к числу людей, которые информировались о применении созданного ими оружия, но гонорары получали исправно, не так ли?

Кулябин машинально кивнул, а потом подумал — ведь он давно подозревал, что «Могиканина» куда-нибудь пристроили, слишком уж регулярно на его банковском счету оказывались деньги от Министерства обороны. Но думать ему об этом почему-то не хотелось...

— Вижу, что так, можете не отвечать, — сказал Андрей, не обративший внимания на почти незаметный кивок Кулябина. — Я все время хотел спросить — а почему вы назвали свое детище «Могиканином»? Это уж слишком отдает детством — игрой в индейцев и прочей книжно-киношной лабудой вроде Гойко Митича.

— Вы же прекрасно понимаете, что такие вещи, как этот проект, не создаются в одиночку. Это просто нереально. Даже для создания какой-нибудь мелочи, которая только и умеет считать, сколько вам жить осталось, исходя из ваших гороскопов — и то нужна помощь одного-двух человек. Хотя бы для того, чтобы увидеть ошибки друг у друга. Так и с этим проектом. Создавали его больше двадцати человек, начиная с физиков, баллистиков, программистов и заканчивая офицерами-безопасниками, отвечающими за секретность проекта. Я отвечал за написание кода, еще двое рассчитывали принципы защиты и нападения, группа из четырех человек прогнозировала боевые задачи, сверяясь с опытом стран, являющихся вероятными противниками... И когда мы собрались в одном очень секретном учреждении в первый раз, чтобы посмотреть друг на друга и узнать, с кем же придется работать плечом к плечу ближайший год или больше, то отвечающий за секретность проекта полковник Бер... В общем, неважно, как его фамилия, короче — он решил познакомиться с нами поближе. И сделал он это очень интересным способом. К нам в кабинет вошла небольшая группа офицеров и обыскала нас и наши вещи. Наверное, на предмет наличия подслушивающей аппаратуры или еще чего-нибудь. А может, для того, чтобы приучить нас к подобному обращению — в дальнейшем подобные процедуры, унижающие человеческое достоинство, выполнялись регулярно. Так вот, у одного из нас в сумке нашли книгу. «Последний из могикан» Фенимора Купера. Он купил ее своему сыну по пути на работу. Полковник взял книгу в руки, пролистал. Потом спросил, помнит ли кто-нибудь из нас, как звали главного героя. Оказалось, что книгу читали многие, почти все, и имя Натаниэля Бампо ни для кого не было секретом. В том числе и его прозвище — Соколиный Глаз. И полковник сказал, что поскольку основой проекта является стопроцентная точность, то имеет смысл назвать его «Могиканин», чтобы отразить меткость главного героя. Многие тогда согласились с ним, опустив глаза и стараясь не смеяться — ведь Могиканином в книге был не Бампо, а Чингачгук, но название пришлось принять — хотя бы из соображений секретности. Догадаться по такому названию о сущности проекта было практически невозможно...

— Вот, значит, как, — покачал головой Андрей. — Даже название оказалось немного с ошибкой...

— Да никакой ошибки! — возразил Кулябин. — Совершенно все правильно! Ракетный комплекс, оснащенный программой боевого патрулирования «Могиканин» стрелял не хуже Чингачгука и Соколиного Глаза вместе взятых! Количество целей, удерживаемых одновременно, превосходило в разы все существующие системы! Скорость ответного удара возросла неимоверно! А обслуживающий персонал можно было сократить не меньше, чем на пять-шесть человек, ибо боевой режим не требовал особых знаний от дежурного офицера!

— Точно, — согласился Андрей. — Мы с Антоненко дежурили по очереди по двенадцать часов, потом капитан от безделья согласился устроить трехсменные вахты. Перешли на восьмичасовый режим. Благо у нас у всех благодаря инъекциям гормональных препаратов день и ночь были искусственно сдвинуты на определенное количество часов, поэтому во время вахты спать никому не хотелось. Мы словно жили в разных часовых поясах. Я, например, по московскому времени, Антоненко — по Камчатке, а капитан Пряхин — черт его знает, по какому, но еще на восемь часов разницы. Программа была простой — хотя та, что была раньше, прежде чем «Могиканина» установили на наши компьютеры, тоже была не очень сложной. Однако она требовала наличия у пульта двух дежурных офицеров, а наше министерство ох как любило сокращать служащих!

— Хотите сказать, что моя... наша программа подвигла военных к мысли о сокращении штата? — Кулябин уже совершенно забыл о пиве, полностью погрузившись в эту беседу и воспоминания.
— Да ничего я не хочу сказать, — Андрей нахмурил брови. — Просто кто-то же должен был ответить... За все то, что произошло.
— Вот мы уже минут тридцать тут стоим, — возмущенно сказал Кулябин, — у меня ни в одном глазу, а вы все ходите вокруг да около, ничего толком сказать не можете, все только обвиняете меня в несуществующих грехах! Лучше бы взяли пива еще и поговорили бы по душам!
— Вам бы все пиво трескать, — презрительно ответил Андрей. — Вы лучше скажите, кто возглавлял проект? Вы?
— А что, вам со мной выпить противно? — проигнорировав вопрос, спросил Кулябин.
— Да вы еще на брудершафт мне предложите! — возмутился Андрей. — Говорите, кто возглавлял создание комплекса? Кто отвечал за его сдачу?
— Я, — кивнул Кулябин, — а кто же еще? На мне был весь код программы. Весь, понимаете?! Я не мог никому доверить такую тонкую вещь! И хотя в нашей команде были талантливые парни, никому из них я не смог дать ни одного мало-мальски значимого задания, так, мелочи всякие, обработчики исключений, интерфейс, прочая фигня... Они, конечно, обижались, пытались на меня воздействовать разными способами. Кто бутылочку поднесет, кто на меня куратору из ФСБ настучит. Меня такие подходы не впечатляли, я добился карт-бланша у руководства и властвовал в группе на правах ответственного программиста.
— Значит, и тестировали ее вы, и отлаживали, и все, что там еще требуется при подготовке — все делали вы?
— Я, — не без гордости сказал Кулябин. — И когда все было готово, упаковал все это чудо, создал универсальный инсталлятор для юниксо-подобных систем, после чего участвовал в первом испытании на секретной базе — свойства программы проверялись и применялись в виртуальной игре с учебно-боевым заданием. Программа, а вместе с ней и ракетный комплекс, чудесно справилась с поставленной задачей. Все цели были уничтожены. И я получил, как вы изволили напомнить мне, довольно приличный гонорар. Но такова судьба всех, кто изобретает что-нибудь совершенное, что-нибудь эксклюзивное и необходимое своей стране. В этом меня обвинять бессмысленно. Ведь вы же, сидя за пультом «Могиканина», не забывали получать денежное довольствие — думаю, очень даже неплохое, исходя из вредности, боевой готовности и секретности. Я немного понимаю в ваших армейских премудростях, так что упрекать меня деньгами не имеет смысла.
— Да, деньги я тоже получал. И вы совершенно правы — неплохие. Но только все кончилось. Очень быстро. С появлением на наших компьютерах вашего гениального творения. Причем для многих кончились не только деньги. Кончилась жизнь.
— Давайте начистоту, — у Кулябина не выдержали нервы. — Рассказывайте, что у вас случилось — а там уже вместе решим, чего, по вашему мнению, я достоин — смерти, жалости или награды. Вы согласны?
Андрей замолчал на пару минут. Его взгляд блуждал где-то высоко в небе; Кулябин чувствовал, что он вспоминает то, о чем хотел рассказать. И казалось, что ничего хорошего эта информация Кулябину не принесет.
— Не знаю, с чего начать, — вдруг сказал Андрей. — Попробую — но за хронологию и эмоции не отвечаю — слишком уж живо все в памяти...

* * * * *

— ...Вчера в Интернете тест нашел, — подойдя откуда-то сзади и заглядывая на экран через плечо, сказал Любашин. — Себя проверил. Все точно. Думаю вот тебя теперь проверить.
— Никогда не подкрадывайся сзади, — раздраженно ответил Антоненко. — Знаешь, ведь я человек нервный...
— А раз нервный — так какого черта ты здесь делаешь? — хохотнул Любашин, присев в соседнее кресло. — У нас ведь работа такая, что нервным и припадочным не место...
— Я не настолько, — оправдался Антоненко. — Не бойся, за работу я в ответе. Вот могу просто в ухо дать, если еще раз так сделаешь.
— Как Маркову? — спросил Антоненко. — Ему-то за что? Он, вроде, к тебе не подкрадывался?
— За дело, — Любашин не отрывался от экрана, следя за вращающимся зеленоватым сектором радара. — Заслужил он...
— А поподробнее?
— А не пошел бы ты, — огрызнулся Любашин. — Что за тест? Вроде собирался меня проверить. До конца смены подождать не можешь?
— Да там особо напрягаться не придется, — Антоненко пропустил резкость напарника мимо ушей. — Пройдешь его, не отрываясь от радара.
— Давай, не томи. Мне скоро меняться...
— Знаю, — усмехнулся Антоненко. — Я же тебя и меняю. Короче — возьми ручку и маленький листок бумаги.
Любашин, по-прежнему глядя строго перед собой на экран и держа в поле зрения сигнализирующие лампы тревоги, протянул руку в сторону, отработанным движением вытащил из ящика стола ручку, пододвинул блокнот и сказал:
— Готов.
— Ну давай... Сейчас посмеемся. Задумай любое число от единицы до девяти.
— А ты подальше отойди, — быстро зыркнув в сторону, потребовал Любашин. — Смотришь во все глаза, что я пишу...
— Да куда ж тут отойти-то? — спросил Антоненко. — За дверь, что ли? Тут же два квадратных метра!
— Спиной встань — мне в экране видно, куда ты смотришь.
— Ладно, недоверчивый ты наш, — Антоненко отвернулся. — Все равно — как ни стой, а ответ... Ну да ладно. Число записал?
— Да.
— Умножь на девять.
— Умножил.
— Теперь в том числе, что получилось, сложи между собой цифры.
— То есть? — Любашин явно был не настроен думать.
— То есть, если получилось двадцать пять, сложи два и пять.
— А-а... — протянул Любашин и выписал какую-то загогулину на бумаге. — Дальше.
— Отними четыре.
— Проще простого.
— То, что получилось, — это буква в алфавите. Отсчитай...
— Сделал, — спустя секунду ответил Любашин. Антоненко усмехнулся, вспомнив, как он сам проходил этот тест — чувствовалось, что напарник на верном пути.
— На эту букву напиши любую страну.
Любашин задумался, постукивая пальцами по столу.
— Чего, тяжко придумать? — спросил Антоненко. Сам он сообразил за пару секунд.
— Не то чтобы тяжко... Придумал, — сказал Любашин и черканул по листку.
— Теперь на третью букву этой страны запиши животное. Любое животное. Сразу говорю — такое животное есть...
— Конечно, есть, — согласно кивнул напарник. — Написал.
— Все написал? И страну, и животное?
— Да. А что, сложности какие-то? Вот со страной пришлось повозиться... А в чем суть?
— Да суть-то в том, Любашин, что в Дании носорогов не бывает.
И тут Любашин все-таки сумел заставить себя на секунду отвлечься от экрана, повернулся к Антоненко и непонимающим тоном спросил:
— Каких носорогов? В какой, нахрен, Дании?
— Ты чего, Любашин? Я же тебе говорю, в Дании носорогов не бывает! Разве ты не это написал?
— Нет, — вернулся к созерцанию радара Любашин.
— А что? Что ты написал?!
— Доминиканская республика. И на букву «эм» — мартышка.
Антоненко не поверил, сделал шаг к столу, взял блокнот, убедился в том, что напарник не врет. Потом шумно втянул воздух носом, хотел что-то сказать, но в итоге молча бросил блокнот назад и вышел.
Любашин пожал плечами.
— Ну, хотел я написать «Дания», чего уж тут скрывать, — сказал он, оставшись в одиночестве. — А там и до носорога недалеко. Но не привык я мыслить стандартно... Не привык, и все. Сразу понял, что тут дело нечисто. Лишь бы он теперь от обиды не забыл, что через пятьдесят минут смена.
Антоненко не забыл, пришел вовремя. Правда, не сказал ни слова, кроме положенной формулы принятия боевого дежурства. Тут уж было не до шуток — все серьезно.
Любашин расписался в журнале, протянул ручку разводящему, который был явно не в курсе теста про носорога; тот подтвер-

дил смену, после чего спросил у Любашина:
— Как будете отдыхать — активно? Или сон? Можем шлем на голову, и в дальние края — только сон закажи.
— Знаю, что можете, — отмахнулся Любашин. — Вы, товарищ капитан, об этом каждый раз спрашиваете. А нас ведь здесь всего шестнадцать человек вместе с вами — привычки каждого вы уже изучили досконально, знаете, что я сначала иду в тренажерный зал, маюсь там с гирями и снарядами до отупения, а только потом — в койку.

Капитан кивнул, соглашаясь, — здесь, на точке, каждый был как на ладони; все знали друг о друге все и даже чуть-чуть больше.
— Могу только добавить к твоему монологу, что нас теперь здесь не шестнадцать, а всего семеро, — покачал головой капитан. — Личный состав сокращен более чем наполовину.
— Основания? — напрягся Любашин, чувствуя, что в их жизнь только что вошли какие-то не очень хорошие перемены.
— На наш сервер сегодня будет установлена новая программа обслуживания ракетного комплекса. Программа не требует большого количества обслуживающего персонала. Дежурства станут проходить в две смены по одному офицеру. Сегодня же у медика всем ответственным пройти смену гормональной терапии, сменить часовые пояса.
— По двенадцать часов — справимся ли? — спросил Любашин, представив себе ту нагрузку на организм, что свалится на них уже в ближайшие сутки.
— Посмотрим, — понимающе кивнул капитан. — На неделю программа будет установлена на резервный компьютер, где пройдет тестирование и будет настроена под нашу конкретную ракету. Будем изучать ее по очереди в свободное от вахты время. Трое суток с нами здесь пробудет один из ее создателей — не самый главный, но более или менее соображающий в ней, после чего будем учиться по книжкам. Через неделю экзамен, после чего еще одна копия будет установлена на главный компьютер.
— Серьезно все, — покачал головой Любашин. — Вы сами смотрели на эту программу? Она в состоянии заменить теперешнее обеспечение?
— Я понимаю, лейтенант, прекрасно все понимаю, — капитан похлопал его по плечу. — Мы все здесь с высшим образованием, дураков нет... Программу писали тоже не лохи. Группа программистов от Министерства обороны, очень засекреченный проект. С тех пор, как наш Президент издал указ — или приказ, — если говорить уж нашим военным языком — о переходе на некоммерческие программы и отказе от Windows, то это наиболее удачный проект. На мой взгляд. Вы, как опытные во всех отношениях компьютерщики, еще сумеете по достоинству оценить его. Тем более, что авторы проекта «Могиканин» — а именно так он и называется, — зная о том, что все мы в одной лодке, в смысле секретности, допуск у нас с вами одинаковый, предоставили вам и мне для изучения исходные коды программы. Предполагается, что ошибок в ней нет, но могут возникнуть какие-нибудь предложения во время работы «Могиканина». И тогда каждый из вас может дописать то, чего ему не хватает. Вот ты, Любашин, явно в состоянии сделать это, о твоих умениях я наслышан. Да и Антоненко не отстанет от тебя, и еще парочка ребят из дежурной службы. По моей рекомендации, здесь после сокращения останутся наиболее талантливые и полезные парни. Вы все распишитесь в очередном приказе о допуске, после чего сможете посмотреть на то, что творится в резервном бункере. Там сейчас работают специалисты, проверяют оборудование на совместимость...

В ТОТ ДЕНЬ, КОГДА ОН ОСТАЛСЯ ОДИН, ОН СОВЕРШИЛ ОШИБКУ. ЧЕРЕЗ ТРИ ЧАСА ПОСЛЕ СКАНДАЛА, ПОСЛУЖИВШЕГО ПРИЧИНОЙ РАЗВОДА. ОШИБКУ, КОТОРУЮ НЕ СМОГ СРАЗУ ЗАМЕТИТЬ И ИСПРАВИТЬ

— А раньше нельзя было проверить? — скривился Антоненко. — Профессионалы, блин. Они что, не знали, для какого железа программу пишут?
— Знали, — сурово посмотрел на него капитан. — Ты вот знаешь, что у тебя в шахте ракета? Знаешь. А какого хрена проверяешь каждый раз, когда на дежурство заступаешь? Ты еще иди потрогай ее, а то мало ли что, вдруг она картонная? «Лейтенант Антоненко пост принял, ракета в состоянии боевой готовности...» В конце добавляй — «Одна штука». Вот и они — есть такая вещь как инструкция. Шаг влево, шаг вправо — расстрел, прыжок на месте — провокация. Поэтому не бухти, а со всем уважением, когда сменишься, иди изучай творение наших Биллов Гейтсов. А пока — очередь Любашина. Давай, лейтенант, дерзай, осваивай новую программу.
— Освою, не переживайте за меня, — пробурчал тот, понимая, что запланированный отдых накрывается по полной программе. — Лишь бы...
— Не понял, — сурово ответил капитан. — Есть какие-то комментарии? Кругом шагом марш!
— Есть! — четко ответил Любашин, подбросив правую руку к несуществующему козырьку, но потом осекся, вспомнив, что на дежурстве они головные уборы не носят с тех пор, как сломался кондиционер. Лихо развернувшись на каблуках, он строевым шагом вышел с поста в коридор.

Уже за дверями он позволил себе расслабиться.
— Вот же принесла нелегкая! — с сожалением произнес он. — А собирался как белый человек, с книжкой расслабиться...

Он уже в «...надцатый» раз перечитывал Толкиена. Книга позволяла максимально отключиться от боевого дежурства, окунуться мир фантазий и отвлечься от мрачных серых казематов, опутывающих пусковую шахту. Так уж получилось, что взял-то он с собой много всяких книг, но в один прекрасный день, во время учебной тревоги по отражению атаки террористов, Антоненко за каким-то хреном включил в жилом отсеке противопожарную систему, хотя коню было понятно, что тревога учебная. Струи воды исхлестали все комнаты персонала и погубили целую книжную полку. То, что у Любашина получилось после сушки, было просто неприятно брать в руки — в системе пожаротушения использовалась какая-то особая жидкость, угнетающая горение, которая превращала все, что может воспламениться, в том числе и бумагу, в нечто, напоминающее труху. Плюс ко всему труха эта страшно, просто нестерпимо воняла. Любашин выкинул все это в утилизатор, с ужасом представляя, что он будет делать оставшиеся восемь месяцев дежурства — и случайно нашел в закрытой наглухо походной сумке толстый том Толкиена, уместивший в себя всю трилогию. Ни мешок, ни его содержимое не пострадали, так как находились в момент тревоги в шкафу, недоступном для брызг убийственной смеси.

Он ухватился за книгу, как за спасательный плотик в этом однообразном мире, который только и состоял из фраз «Дежурство сдал — дежурство принял» и мрачно-зеленоватого экрана радара, сканирующего округу в несколько десятков километров. Иногда, правда, приходили письма от родных — в последнее время все реже и реже. Сестра вышла замуж и уехала к черту на кулички, ей было не до затерявшегося в недрах секретной пусковой установки брата. Отца почти не помнил, мать — ждет его каждый год в отпуск, надеется, что в очередной раз приедет к ней с молодой женой. Одного не понимает — где же ее взять-то здесь, эту самую молодую жену? Он даже толком не знает координат шахты, кругом одни секреты; вокруг на много километров тайга, а маме приходится писать совершенно другие вещи, поскольку военную цензуру еще никто не отменял. Да и очень было неприятно знать, что твои конверты вскрывает суровый, бездушный офицер ФСБ, читает написанные маме строки, пытается выхватить в них признаки государственной измены, да все никак не получается, и его за это не повышают в звании и не продвигают по службе, и он от этого злится, нервничает, пьет и порой выкидывает в мусор чужие мысли, чужие пожелания, страхи и радости, чтобы хоть как-то досадить этому миру...

Все это Любашин думал, медленно идя по коридору в сторону запасного командного пункта. Думал, переживал, а потом решил, что освоение новой программы — это ведь тоже своего рода развлечение, надо отнестись к этому, как к подарку судьбы. Не прочитаем Толкиена, так хоть узнаем, как отражать атаку превосходящих сил противника!

Он вошел в комнату, которая обычно пустовала в силу того, что основной пост пока ни разу за все время существования шахты

не отказал. Работал себе, потихоньку оглядывая горизонт, шевеля чашами локаторов, спрятанных под маскировочной сеткой, просчитывая варианты возможных траекторий с учетом расположения стационарных шахт противника и перемещения мобильных установок по данным разведки. Вот только решения принимать еще ни разу не доводилось — и слава Богу. Режим боевого дежурства протекал спокойно, слаженно, без каких-либо форс-мажорных обстоятельств; никто не собирался атаковать нашу страну, ничьи ракеты не обнаруживали бесконечно работающий радар.

Войны не было. И замечательно...

Любашин, войдя на пост, увидел там двух человек, о чем-то оживленно беседующих за экраном компьютера. Сам экран появился здесь только сегодня — вмонтированный прямо в крышку стола, словно он лежал прямо перед тем, кто принимал бы дежурство. Поверх монитора была какая-то тонкая прозрачная то ли пленка, то ли лист пластмассы...

— Панель сенсорная, так что не влезьте в нее раньше времени, — предупредил один из компьютерщиков, будто угадав мысли Любашина. — Управление ведется стилом прямо на экране. Необходимость ввода команд с клавиатуры минимальна, в основном для обслуживания системы. На боевом дежурстве ваш первый друг и товарищ — вот эта панель.

Любашин подошел поближе, заглянул через плечо того, кто сидел за столом. Действительно, все отличалось от того, что они привыкли видеть. Экран, поделенный на сектора, большие цифры с комментариями — полная расшифровка информации, поступающей с радара, анализ целей (в настоящий момент отключенный), еще нечто, не сразу доступное пониманию... И в правом нижнем углу экрана большая красная кнопка, похожая на стеклянную (постарались разработчики интерфейса!).

«ПУСК».

— Да... — покачал головой Любашин. — А если я в нее локтем спросонья...

— Не выйдет, — ответил инженер. — Необходимо ввести голосовой пароль. Каждый из вас потом пройдет процедуру записи, анализатор голоса запомнит вас и ваш пароль — никто из вас не в свою смену не сможет произвести старт ракеты. Плюс — прямо как в персональном компьютере...

— Это как? — заинтересованно спросил Любашин.

— Будет дополнительный вопрос. Что-то типа: «Вы на самом деле хотите произвести пуск?» После чего необходимо будет назвать второй подтверждающий пароль. Мы проверяли — эта процедура проходит за три секунды, при желании можете отрабатывать ее до совершенства, сократите до двух.

— Вы знаете, сколько пролетит стратегическая ракета за две секунды?

— Знаю, не думайте, что мы дилетанты. Это допустимая погрешность при отражении атаки. Нельзя все доверить машинам, даже несмотря на то, что время их реакции сэкономит именно эти секунды.

Любашин кивнул (все мы в детстве насмотрелись «Терминаторов»). Потом взял в руки книгу с очень привычным названием «Инструкция» и углубился в чтение, периодически сверяясь с экраном. Оказалось не так уж и сложно — теорию Любашин всегда усваивал легко. Пролистав около двадцати страниц, он уже неплохо ориентировался в показаниях дисплея и даже понимал (не все, правда), о чем говорят инженеры, тестируя программу.

Спустя пару часов один из компьютерщиков махнул рукой и сказал:

— Все, хватит. Похоже, параметры программы в норме. Все исходные данные внесены, учтены климатические условия, сейсмологическая активность в вашем районе и еще куча параметров, о которых даже не стоит задумываться. Программа в вашем распоряжении на одну неделю. Несколько часов чистого времени, потраченного на изучение программы, будет достаточно для того, чтобы заступить на вахту. Надеюсь, инструкция написана на понятном языке?

— Да уж, — хмыкнул Любашин. — Давайте, где мне расписаться?

— Смотри-ка, опытный, — улыбнулся второй инженер. — Знает, что просто так такие книжки в руки не дают. На, вот ручка, вот бумага...

Любашин просмотрел большой бланк с множеством пунктов и подпунктов, отметил про себя внизу подпись «Руководитель проекта Кулябин Д.А.», расписался там, где ему показали, и проводил взглядом уходящих инженеров. Парни оставили очень приятное впечатление слаженностью своей работы, профессионализмом и отсутствием того, что в обыденной жизни называлось «понты». Этим у них и не пахло. Они прекрасно понимали, что имеют дело с таким же знающим человеком, как и они сами, — недаром коды проекта были переданы персоналу шахты для изучения.

Подумав об этом, Любашин полностью погрузился в изучение нового программного обеспечения; этот процесс увлек его и всех офицеров, допущенных до вахты, на ближайшую неделю. Они с честью вышли из этого испытания, сдав все тесты по отражению виртуальной атаки вероятного противника на «отлично». Впрочем, они всегда были отличниками боевой и политической...

А ровно через неделю их пусковая установка была на три часа снята с боевого дежурства, зона, за которую отвечала их шахта, передана для слежения, эскадрилье ракетоносцев, а та же самая пара молодых, но опытных инженеров загрузила на основной компьютер «Могиканина». Программа протестировала новые условия обитания и осталась довольна тем железом, которое было теперь для нее базовым. Капитан, в тот день узнавший, что приказ о присвоении ему очередного звания подписан Министром обороны в связи с освоением новой техники, и в ознаменование очередного праздника рода войск, к которым он не имел никакого отношения, тихо радовался, собираясь вечером выпить рюмку коньяка из старых запасов...

Но выпить не удалось.

Потому что ровно через сорок минут после установки «Могиканина» случилось то, что потом назвали «северным инцидентом».

Антоненко, принявший к этому времени ночную вахту (которая была для него днем по московскому времени), в двадцать три часа четырнадцать минут зафиксировал нарушение границы Российского государства воздушным судном, опознанным «Могиканином» как бомбардировщик Б-52.

Предположительно с ядерным боезапасом на борту...

— Есть контакт с целью! — передал он капитану. — Цель — воздушная, ядерный бомбардировщик, высота предельно малая! Дальность семьдесят километров! Курс юго-запад, строго по прямой!

«Могиканин» мигал на экране столбиками цифр, уточняя все данные по цели. Капитан ворвался на пост, как вихрь.

— Доложить командующему округом! — сказал он сам себе и схватил со стены телефонную трубку прямой секретной связи. — Срочно «Лидер»!

Барышня на том конце провода прекрасно знала свою работу. Знала, что в любое время по этой линии может пройти очень и очень срочный звонок. Поэтому застать ее врасплох было невозможно.

Гудок, легкое потрескивание в трубке. Антоненко продолжал комментировать продвижение самолета в российском небе.

— Товарищ генерал-лейтенант, докладывает майор Лукьянов! — просто кричал в рубку капитан, забыв о том, что звезд на его погонах пока не хватает. — Сигнал «Свежий ветер»!

«Какая чушь, — подумал Антоненко, нервно постукивая пальцами по столу и не отрывая взгляда от экрана, где с быстротой скорости света сменялись данные телеметрии. — Сигнал «Дирол — морозная свежесть»! Кто всю эту фигню придумывает?! А еще лучше — «Несвежий оливье»!»

Краем уха он пытался услышать, что же там бурчит в трубке командующий. Понять ничего было нельзя, но капитан (внезапно окрестивший себя майором) часто кивал, после чего коротко доложил ту информацию с экрана, которая выводилась ему на дублирующий монитор прямо к телефону.

Тем временем самолет продолжал двигаться вглубь территории России. Антоненко на пару секунд отвел глаза в сторону, проследил за электронным глобусом и определил приблизительно шесть или семь крупных городов, которые накрыло было ядерным взрывом, рискни Б-52 сбросить бомбу сейчас.

ЛЕЙТЕНАНТ АНТОНЕНКО ПОСТ ПРИНЯЛ, РАКЕТА В СОСТОЯНИИ БОЕВОЙ ГОТОВНОСТИ...

— Не может быть... — внезапно прошептал капитан. — Товарищ гене... Я смотрю на экран — цель опознана, захвачена, сопровождается!

Трубка что-то коротко буркнула. Разговор окончился.

Капитан приткнул трубку обратно на стену, непонимающим взглядом посмотрел на Антоненко, на глобус, на радар и сказал:

— Их спутники не подтверждают информацию «Могиканина». Самолета, по их мнению, не существует. Приказано — цель со-

провождать, однако провести тестирование на втором комплекте программы на запасном командном пункте...
— Они не верят?
— Дело не в вере, — посмотрел на него Лукьянов. — Мы не отвечаем за наш сектор в одиночку, и кому, как не тебе, это понимать. Их техника ничего не видит. А по курсу самолета, сам видишь — шесть городов.
— Семь, — машинально уточнил Антоненко. В этот момент на пост вошел Любашин.
— Будем стрелять? — сразу же спросил он. — Высота позволяет завалить его прямо в тайгу до подлета к городам.
— Откуда знаешь?
— На запасном проверил.
— И там тоже есть цель? — спросил капитан.
— А что, есть сомнения?
— Есть, да еще какие... — Антоненко не скрывал своего волнения. Никто не мог точно сказать, к чему приведет вся эта чехарда с целями — вполне могло кончиться Третьей мировой войной, в которой им жить бы осталось несколько часов, до первой волны вражеских стратегических ракет.
— Кто сомневается?
— Командование. Причем не просто на уровне «Чей самолет, зачем летит?» Все куда серьезнее — есть самолет или нет?
— Фантом? — поднял брови Любашин. — Сгенерированный «Могиканином»? Думаете, такое возможно? Нам поставили программный комплекс, который сам создает цели? Но мы же не игровой салон где-то в провинциальном городке, мы здесь не в Counter-Strike играем!
— Объект углубился на территорию России на восемьдесят километров, — тем временем сообщил Антоненко. — Жду приказа.

Ждал не только он. Ждали все.

Ситуация была, что называется, на грани.

Каждый из них вспомнил в эти минуты все, что только помнил о нарушениях государственной границы: и сбитый южнокорейский самолет, и Матиаса Руста, приземлившегося на Красной площади, и ракету, отправившую на дно Черного моря во время учений украинской армии самолет с мирными пассажирами... Командование сделало их крайними, предоставив принять решение на месте.

Капитан попытался еще раз связаться с командующим округом, но никаких новых указаний не получил — спутниковая служба слежения самолет не видела, вторжения не подтверждала. А экран «Могиканина» мигал предупредительными лампами и отсчитывал километры, оставшиеся до атаки на ближайший районный центр.

Любашин и Антоненко смотрели на своего командира и ждали его решения. Так уж повелось в армии — старший офицер в ответе за все.

Лукьянов протянул руку к трубке секретной связи и сказал командующему:
— Исходя из сложившейся ситуации, принимаю решение — атаковать и уничтожить цель. Средства — ракета «Земля-воздух». Время — текущее.

И не дослушав, что ему там кричал генерал, положил трубку и шепнул себе:
— И да простит меня Бог...

А потом скомандовал:
— Ракете — пуск!

Антоненко произнес свой голосовой пароль, прикоснулся на экране к стеклянной кнопке старта и продублировал команду. Стены слегка дрогнули — наверху откатился в сторону бронелюк, скрывавший горловину шахты. Сквозь толстые бетонные перекрытия они ощутили старт, как громкое шипение. Спустя несколько секунд все стихло.
— До контакта с целью две минуты, — прокомментировал Антоненко, не отрываясь от таймеров «Могиканина». Цель держит точно, идут встречными курсами. Цель маневров уклонения не предпринимает...

ПОЧЕМУ ВАС ОСТАВИЛИ В ЖИВЫХ ПОСЛЕ ВСЕГО, ЧТО МНЕ ДОВЕЛОСЬ ПЕРЕЖИТЬ

* * * * *

Андрей замолчал. Кулябин слушал его, затаив дыхание.
— Что? Что было дальше?! — не выдержал он возникшей паузы. — Вы попали? Попали или нет? «Могиканин» отразил нападение?

Любашин отрицательно покачал головой. Чувствовалось, что он вообще ничего больше не хочет говорить. Словно дальше в этой истории была какая-то чудовищная правда, которую не стоило произносить вслух лишний раз. Кулябин дернул его за рукав камуфляжа.

Андрей вздрогнул и отшатнулся, будто ему были противны прикосновения подвыпившего программиста.
— Дальше? — переспросил он. — Попали? Послушайте, я ведь до сих пор не верю в то, что случилось. И мне по-человечески интересно, почему вас оставили в живых после всего, что мне довелось пережить.
— Неужели... Неужели программа... Что там случилось? — схватился за голову Кулябин. — Я ведь ничего, совсем ничего не знаю!
— Все дело в том, что мы не попали, — медленно проговорил Андрей. — Потому что некуда было попадать. Когда ракета достигла — в кавычках — несуществующего самолета, отметка о бомбардировщике исчезла. И когда Антоненко сообразил, что ракета летит дальше — было уже поздно. «Могиканин» автоматически включил самоликвидатор. И наше титановое чудовище полностью уничтожило большой таежный поселок с населением в тысячу двести человек...
— Какой ужас... — побледнел Кулябин. — А самолет? Куда он делся? И откуда взялся?!

Любашин помолчал немного, потом ответил:
— Спустя пару секунд после уничтожения ракеты на экране появилось сообщение. Я помню его дословно: «Виртуальная цель поражена. Режим обучения переходит на следующий уровень». А когда Лукьянов потребовал от Антоненко зафиксировать все то, что произошло, следующее сообщение повергло всех в шок. Не думаю, что вы сами в состоянии понять, что сделали...
— Говорите, не скрывайте от меня ничего... — дрожащим голосом попросил Кулябин, пытаясь представить, что же могло быть такого в «Могиканине», за что он заслуживал смерти.
— При попытке сохранения информации программа сообщила, что находится в режиме демо-версии и не может записать данные на диск. А потом попросила заплатить Министерству обороны восемьсот пятьдесят тысяч долларов и зарегистрироваться через интернет... Я ведь понимаю, что это шутка, Дмитрий Анатольевич... Но почему вы не отключили ее, Кулябин? Почему?! Мы сожгли поселок с людьми, потом отряд спецслужб пришел к нам, потому что надо было спрятать за семью печатями тех, кто устроил все это, и расстрелял весь личный состав пусковой установки прямо в тайге, возле каких-то огромных муравейников, и я сомневаюсь, что от этих несчастных людей осталось хоть что-нибудь, кроме костей в течение пары дней!!! И лишь я чудом уцелел — мне попали в грудь и шею, но контрольного выстрела не сделали, и я уполз... Восемь дней в тайге, потом золотой прииск, какие-то шаманы, перевязки с зельями, дым от костров, тучи мошек... Я видел тот поселок, точнее сказать, то, что от него осталось. Сожженные дома, вываленный лес. Братская могила. Сам я официально мертв. А вот вы — вы живы. Почему?
— Меня попросили... Чтобы там была такая вот... Шутка. Но она, — голос Кулябина хрипел, — она должна была быть по умолчанию отключена... Я забыл... Я просто забыл...
— Но как? — непонимающе спросил Андрей Любашин. — Как можно забыть ТАКОЕ?
— В тот день, когда я сдавал проект, я поругался с женой... — вспомнил Кулябин тот летний день и трижды проклятый аккумулятор. — Поругался... Вдрызг. И ушел от нее. Понимаете, я был на взводе, плохо соображал... Но ведь все остальное прекрасно работало! — внезапно возмутился он. — Все работало идеально! Программа превосходит все заграничные аналоги!..

Любашин смотрел на Кулябина, не в силах ничего сказать.
— Я виноват, — едва не кричал тот. — Виноват, я же не специально!..
— Да, — тихо ответил Любашин. — Да, конечно...

Потом он повернулся к нему спиной и ушел. Уж очень сильно ныла старая рана на шее...

Он шел и думал, как просто и одновременно сложно устроен мир. И вспоминал своих друзей, которые умерли, потому что кто-то поссорился с женой **С**

ИСХОДНИКИ ВСЕЛЕННОЙ

КОЛОНКА КРИСА КАСПЕРСКИ

— DO YOU BELIEVE IN FATE, NEO?
— NO, 'CAUSE I DON'T LIKE THE IDEA, THAT I'M NOT IN CONTROL OF MY LIFE.
MATRIX

СВОБОДА ВОЛИ ИЛИ ПРЕДОПРЕДЕЛЕННОСТЬ?

➔ **введение.** Каждый из нас, так или иначе, задумывается: насколько все предопределено, существует ли судьба, и можно ли ее обойти? Фатализм (то есть вера в неизбежность) очень удобен нытикам, перекладывающим ответственность за все промахи и неудачи на злодейку-судьбу, но большинство здравомыслящих людей все-таки хотят управлять свой жизнью самостоятельно. Субъективно мы ощущаем свободу выбора: захотим выпить пива — и выпьем: тут никакой фатализм нам не указ. Но вдруг эта «свобода» всего лишь иллюзия? Разве можно доверять своим чувствам? И ведь даже не органам чувств, а... категориям сознания? Может быть, нам только кажется, что у нас есть выбор, а на самом деле все выбрано задолго до нас? Может, это только иллюзия выбора? Потому что на самом-то деле выбора нет: ты открываешь новую банку пива, поднимаешь правую или левую руку, и только тебе кажется, что ты свободно выбираешь, что тебе выпить/дунуть/поднять. Иллюзия — это когда тебе кажется, что что-то есть, а на самом деле этого нет. Тебе кажется, что ты свободно выбираешь, а на самом деле твой выбор предопределен. «...You didn't come here to make the choice. You've already made it. You're here to try to understand why you made it» (c) Matrix Reload.

Существует только один надежный способ проверить, существует ли предопределенность или нет — сгонять в будущее и подсмотреть, а потом вернуться и поступить наоборот. Но машин времени пока что не существует, а обращение ко всяким предсказателям и гадалкам (даже если допустить, что среди них встречаются и настоящие) еще не дает ответа на вопрос: можно ли, зная будущее, его изменить? И если да, то какое же тогда будущее предсказывают гадалки? «What's really going to bake your noodle later on is ... would you still have broken it if I hadn't said anything?» Опять Матрица! Ох, и не простой это фильм, поднимающий серьезные философские вопросы, скажу я. Разбил бы Нео вазу, если бы Пифия ничего не сказала? Настоящие (да и ненастоящие) предсказатели в значительной степени формируют будущее, а не предвидят его (нельзя предвидеть стул в своей комнате, его можно либо видеть, либо нет — то же самое относится ко всем «предвидящим будущее»).

Но все-таки, в чьих руках находится наша судьба? Наука не может дать ответа на этот вопрос, но, быть может, нам помогут религия и философия?

➔ **свобода выбора с религиозной точки зрения.** В подавляющем большинстве религий в том или ином виде постулируются два тезиса: Бог даровал человеку свободу выбора или, по-английски, free will (A), но в то же самое время ни один волос без его воли не упадет с ни с чьей головы (Б). На первый взгляд, эти тезисы кажутся противоречивыми и взаимоисключающими друг друга, однако, на самом деле, никакого противоречия здесь нет, и оно возникает лишь при попытке интерпретации священных книг посредством логики, несостоятельность которой демонстрирует куча парадоксов, придуманных еще в Древней Греции. Подробнее об этом можно прочесть на сервере еврейского культурно-религиозного центра «МАХАНАИМ»: **http://www.machanaim.org/philosof/phil/ph6.htm**. Почему я выбрал именно еврейский центр? Да потому, что иудаизм равноудален от господствующих религиозных течений — ислама и христианства, хотя они в определенной степени являются своеобразной «надстройкой» над ним. Так что лучше опираться на фундамент, а не на крышу. Крыша же — очевидный намек на буддизм, находящийся в довольно интересных взаимоотношениях со свободой воли и предопределенностью.

Традиционная культура Китая издавна передавала людям важные концепции и принципы, такие как Небо, Дао, Бог, Будда, судьба, пред-определенность и т. д. Мудрецы говорили: «Мир непостоянен, все рождается и умирает, начинается и заканчивается, день сменяется ночью, лето — зимой, тишина — звуком, жара — холодом, и нет тому исключений нигде. В мире все взаимозависимо: одно проистекает из другого, другое порождает третье и т. д. Все существующее имеет причину. Состояние любого феномена определяется набором причин, которые привели его в это состояние. Камень падает на землю — причиной является сила притяжения. Причина появления цветка — семя, брошенное в землю». Резюмирую: состояние всего нашего мира определено набором причин, которые привели его к этой кондиции.

Взаимозависимость всего во всем — это и есть карма. Каждый человек на земле имеет разные изначальные возможности, рождаясь тем, кем ему суждено или предписано было родиться. Непостоянство и карма порождает страдание. Вопрос: «А что остается?» Свобода воли. Но не является ли состояние нашего «Я», нашего ума, нашей воли тем же самым набором причин, которые привели нас к этому состоянию? Что определяет наш выбор, наши мысли и поступки? Разве не предшествующее настоящему моменту состояние нашего сознания?

Ответ: в буддизме существует понятие Колеса Бытия или Сансары, в котором происходит бесконечное блуждание сознания (или, выражаясь христианскими терминами, — души). Будда своим опытом указал на возможность выхода из круга рождения и смерти или, иными словами, освобождения, которому сначала предшествует просветление или пробуждение сознания от иллюзорного восприятия окружающего мира. Человек, идущий путем просветления или Дхармы, постепенно уходит из-под влияния кармы, исчерпывая ее и изменяя таким образом свою судьбу (подробнее см: **http://buddizm.sexnarod.ru/topic86589.html** и **universalinternetlibrary.ru/forum/index.php?s=33280cbb0896819b91e35d23dfd0a352&showtopic=912**).

Кстати, ты заметил разницу между буддизмом и иудаизмом? Если иудаизм постулирует свободу выбора (типа, так сказал Всевышний), то буддизм, наблюдет за окружающим миром, обобщает свой опыт и путем аналогий приходит к детерминизму, который тут же элегантно обходит путем ментальных методик.

С точки же зрения индуизма, весь мир представляет из себя сон Брахмы, спящего на поверхности молочного океана на далекой планете. Наша вечность — только секунда для него. Когда Брахма проснется, мир для нас закончится. Естественно, никакой свободы воли у персонажей чужого сна нет, как, впрочем, нет и предопределенности (сны всегда непостоянны и хаотичны, даже если это сны Брахмы).

Таким образом, обобщенный религиозный опыт не дает ответа на наш вопрос, оставляя его предметом веры, каждая из которых рассказывает свою собственную сказку.

➔ **свобода выбора с научной точки зрения.** Во времена Ньютона существовало убеждение, что миром правят законы (термодинамики, например), которые не в силах нарушить даже сам Бог! Собственно, для Ньютона Бог и был монархом-законотворцем, устанавливающим законы природы и бдительно контролирующим их исполнение. Тут, кстати, можно было бы подискутировать на тему, что представляет собой Бог — личность или машину? В первом случае Бог может нарушать установленные им же законы, во втором — нет. Но лучше отложить этот разговор на следующий раз, а то нас уже и так заносит.

Убеждение это носило скорее религиозно-догматичный, чем научный характер, и не было ничем обосновано или подкреплено. Результаты экспериментов никогда в точности не соответствовали теории и, кроме того, если какой-то закон срабатывал 1000000000 раз, у нас нет никаких гарантий, что то же самое произойдет и в 1000000001. Ученые знали, что все законы имеют границы применимости. В частности, расхождение в наблюдениях за Меркурием с теорией Ньютона, ничего не «знающей» об искривлениях пространства-времени, привели к созданию новой теории относительности, а попытки подсчета полной энергии излучения черного тела, которая в классической физике равнялась бесконечности, заложили первый камень в храм квантовой физики, но к квантовой физике мы еще вернемся, а пока продолжим говорить о ньютоновских временах.

Нам, жителям XXI века, избалованным наукой и прогрессом, трудно (если вообще возможно) представить, какой восторг вызвал расчет Эдмунда Галлея (не путать с Галилеем) времени появления кометы, названной впоследствии его именем (кстати говоря, для этого Галлею требовалось решить систему дифференциальных уравнений, чего он сделать не смог, и обратился за помощью к Ньютону, у которого такое решение было, но только тот не придавал ему большого значения, и именно Галлей убедил его опубликовать свой труд, ныне известный каждому школьнику). Окрыленные успехом, ученые поверили в то, что, опираясь на законы физики, они могут прогнозировать события на сколь угодно длительный срок, главное — учесть все факторы (типа, трение, сопротивление воздуха и др.) и произвести тщательный расчет.

Пьер Симон Лаплас в «Опыте философии теории вероятности» описал любопытную концепцию (сейчас известную под именем «Демона Лапласа»), из которой следовало, что, зная координаты и импульс каждой частицы Вселенной и имея в своем распоряжении достаточно мощный вычислитель, мы сможем рассчитать абсолютно все события, которые только произойдут. В дословном переводе это звучало так: «Ум, которому были бы известны для какого-либо данного момента все силы, одушевляющие природу, и относительное положение всех ее составных частей, если бы вдобавок он оказался достаточно обширным, чтобы подчинить эти данные анализу, обнял бы в одной формуле движение величайших тел Вселенной наравне с движениями легчайших атомов: не оставалось бы ничего, что было бы для него недостоверно, и будущее, как и прошедшее, предстало бы перед его взором», из чего следовал полный, абсолютный детерминизм и фатализм.

Никакой свободной воли! Ну, разве что только объявить существование нематериальной «души», не подчиняющейся законам физики, но... это нечестно и антинаучно. Ни в одном эксперименте ничего похожего на душу зафиксировано не было, а наличие предполагаемой свободы выбора является всего лишь субъективным ощущением, которое вполне может быть иллюзорным. В конечном счете, наше сознание — это продукт биохимических процессов, подчиняющихся все тем же физическими законами, без признаков нарушения последних. Впрочем, даже не прибегая к физике, легко показать, что поступки человека в значительной мере диктуются его взаимодействием с окружающими людьми и грамотным психологом легко прогнозируются. Весь вопрос в том, с какой точностью выполняется прогноз.

А вот с точностью дела всегда обстояли туго. Если в нашей воображаемой вселенной имеются всего два тела, вращающиеся вокруг друг друга, то их положение легко рассчитывается на любой момент времени t с бесконечной точностью. А вот с тремя телами уже возникает серьезная проблема. Адекватного математического аппарата у нас нет, поэтому приходится прибегать к дифференцированию, то есть «расчленять» орбиту на короткие отрезки, рассчитывая траекторию шаг за шагом. Чем короче отрезки, тем выше точность, и дольше расчет, но как бы там ни было, с течением времени ошибки вычислений стремительно нарастают, расчетная орбита все больше отклоняется от реальной, и в вычислениях возникает неопределенность. Теоретически (чисто теоретически), имея неограниченно мощный вычислитель (или адекватный математический аппарат), эту проблемы было бы можно обойти, но практически...

Практически дела обстоят так. В равновесных системах законы физики очень даже нехило рулят. Шарик, покоящийся в ложбине, занимает устойчивое положение и к незначительным возмущениям/погрешностям в расчетах нечувствителен. Шарик же, помещенный на гребень волны (карандаш, поставленный на острие), находится в крайне неустойчивом состоянии и остро реагирует на малейшие возмущения, нарастающие со скоростью снежной лавины.

Вся соль в том, что значительная часть объектов нашей вселенной представляет собой неравновесные системы, взять хотя бы Земную атмосферу, например. Прогноз погоды на длительный срок невозможен потому, что малые возмущения ведут к большим возмущениям, не заложенным в климатическую модель, но радикально меняющим облачность на цунами.

Теперь самое время вернуться к квантовой физике, наглядно демонстрирующей, что поведение частиц в микромире не определено наперед. Рассмотрим такое известное явление, как период полураспада. Если период полураспада такого-то атома составляет 100 лет, то это означает, что если взять XXL атомов, то через 100 лет XXL/2 из них распадутся, но если взять один атом, то ничего определенного о его судьбе сказать нельзя. Он может распасться как через минуту, так и через миллиард лет. Означает ли это, что «одинаковые» атомы вовсе не так одинаковы, как это кажется нам? Имеют ли они сложную внутреннюю структуру, до которой мы еще не добрались, или все-таки распадом атома управляет именно вероятность?

Квантовая физика вызывала отвращение у Эйнштейна, который до конца жизни повторял: «Бог не играет в кости», но... авторитет Эйнштейна — это одно, а реальные наблюдения — совсем другое. Самым значительным открытием конца XX — начала XXI века стало осознание того факта, что никаких законов вообще нет, и природой правит один лишь Хаос. Другими словами, все без исключения законы имеют вероятностную природу. Взять хотя бы закон сохранения энергии, который «работает» по принципу орел-решка, то есть при большом числе подбрасываний «честной» монетки мы получим 50 на 50, но вот на малых... на малых временных отрезках наблюдаются флуктуации. Вечный двигатель возможен, только очень маловероятен. Возьмем обычную термопару и подключим к ней нагрузку. Поскольку молекулы воздуха движутся хаотично, то рано или поздно наступит момент, когда возле одного конца термопары соберется больше медленных молекул, а возле другого — быстрых, и тогда термопара начнет выдавать ток. Если запастись терпением — можно дождаться флуктуации, которая будет длиться сто, тысячу или даже миллиард лет, а то и всю вечность. Шум в ПЗС-матрицах имеет ту же самую природу, и фактически матрица представляет собой вечный двигатель, только очень маломощный.

➔ **заключение, или каков мир в действительности.** Вселенная представляет собой совокупность хаотичных самоорганизующихся систем. Равновесные системы всецело подчиняются законам физики и полностью детерминированы. Поведение же неравновесных систем (к числу которых относится и человеческий мозг) может быть спрогнозировано только на какой-то конечный (и притом очень небольшой) срок, да и то с оговорками на вероятность.

Следовательно, предопределенности нет, а вот с судьбой еще предстоит разобраться. Как было сказано выше, единственная возможность выяснить это — заглянуть в будущее. Физика всего лишь открывает лазейку, позволяющую (теоретически) описать вселенную, в которой нет судьбы, но вовсе не отрицает возможность существование некой силы, которая всеми нами управляет, поэтому вопрос остается открытым...

Кстати говоря, часто приходится слышать нелепое утверждение: что, мол, если все предопределено наперед, то давайте отменим ответственность за преступления (типа, я не виноват, что украл/бил/изнасиловал, — это карма такая). Помилуйте! Если все предопределено, то и ответственность предопределена тоже. Если мы в состоянии решить, отменять ее или не отменять — следовательно, свобода выбора все-таки существует, а раз существует свобода выбора, тогда должна существовать и ответственность ©

Попробуйте подписаться в редакции, позвоните нам.

(это удобнее, чем принято думать :)

SYNC

Лучшие цифровые камеры


Хакер

Хакер Спец

Железо

Страна Игр

PC Игры

Мобильные компьютеры

Maxi Tuning

Total DVD

DVD Эксперт

Total Football

Onboard

Mountain Bike Action

Хулиган

Свой бизнес

- Для подписчиков в Москве курьерская доставка **БЕСПЛАТНО** в день выхода журнала
- Дешевле, чем в розницу
- Гарантия доставки и замены в случае потери
- Специальные предложения для подписчиков
- Первый номер подписки высылается по звонку вместе с заполненной квитанцией для оплаты

8-495-780-88-29 (для Москвы)
8-800-200-3-999 (для России)
ВСЕ ЗВОНКИ БЕСПЛАТНЫЕ
Мы работаем с 9 до 18 по рабочим дням

(game)land

"ФУТБОЛЬНЫЙ МЕНЕДЖЕР"!

СОЗДАЙ СВОЮ КОМАНДУ ИЗ РЕАЛЬНЫХ ИГРОКОВ И ПРИВЕДИ ЕЕ К ПОБЕДЕ

ТЫ ПОЛУЧАЕШЬ $135 МИЛЛИОНОВ
на приобретение игроков российской премьер-лиги при регистрации на сайте www.total-football.ru.
Подробности на сайте **www.total-football.ru**

ГЛАВНЫЙ ПРИЗ – ПОЕЗДКА НА ФИНАЛ ЛИГИ ЧЕМПИОНОВ 2006/07

СИТИ-БРЕЙК

отдых по индивидуальному проекту

< ГОРОД
< РОМАНТИКА
< МУЗЕИ
< НА ВОДЕ
< ЭКСТРИМ
< ТУСОВКИ
< ЭКЗОТИКА

ГОРОД

Москва
www.luzhniki.ru

Открытый бассейн

Москва
www.aka-te.ru

Школа суши

Россия
orcinus.ru/dolphinariums/
Москва
Московский дельфинарий
www.nice.pp.ru

Дельфинарий

Фото: IMAGEBANK./FOTOBANK.COM

НА ПРИРОДЕ НЕТ ПЛОХОЙ ПОГОДЫ: КЛЕЙКИЕ ЛИСТОЧКИ, ПРЯНЫЙ ЗАПАХ ГРИБОВ, ИСКОРКИ НА СНЕГУ. В ГОРОДЕ ЖЕ ЛЮБАЯ ПОГОДА — НЕ ПРИВЕДИ ГОСПОДЬ. ПАРУ МЕСЯЦЕВ — ОСТЕРВЕНЕЛЫЕ МОРОЗЫ, ЕЩЕ ВОСЕМЬ — СЛЯКОТЬ С ВМЕРЗШИМИ В НЕЕ ПУСТЫМИ БУТЫЛКАМИ, А В КАЧЕСТВЕ ЛЕТА — ГОРЯЧАЯ ПЫЛЬ, ТОПОЛИНЫЙ ПУХ И ПЛАЧУЩИЕ ОТ ЖАРЫ КОНДИШНЫ. НО, ДАЖЕ ПРИ ТАКИХ ИСХОДНЫХ, ЛЕТО — ЭТО ХОРОШО.

ОТКРЫТЫЙ БАССЕЙН
Ждать милостей от природы, тем более среднерусской, бесполезно: лазоревый океан в центре города все равно не вырастет. Зато с недавних пор там стали появляться открытые бассейны, возле которых можно расслабиться и позагорать. Неоспоримые преимущества по сравнению с океаном — отсутствие акул и шаговая доступность. Абонемент на день — 700 рублей.

ШКОЛА СУШИ Суши, несомненно, входят в золотой фонд мировой кулинарии и почти наверняка — в сферу ваших личных интересов — вкусно, полезно и несложно в приготовлении. В школе шеф-повара ведущих японских ресторанов Москвы научат готовить суши, сашими, роллы, супы, темпуру, терияки. Однодневный курс — 4500 рублей. На сайте школы есть также и виртуальный курс.

ДЕЛЬФИНАРИЙ Человек человеку волк, при самом благоприятном стечении обстоятельств — коллега. Дельфины же относятся к себе подобным и своим старшим братьям по млекопитанию гораздо позитивнее. В некоторых дельфинариях разрешают поплавать с их питомцами: группа состоит из 4–8 человек и 2–3 дельфинов. Единственное условие участия в аттракционе — умение плавать без вспомогательных средств и помощи друга. Часовой сеанс межвидового общения облегчит ваш кошелек на 3900 рублей.

Аксессуары

1. Yanga!
Все цвета музыки!
Yanga! — новый легальный онлайн-магазин музыки с уникальным каталогом, удобной и безопасной системой оплаты и дружественным интерфейсом. Каталог пополняется еженедельно и скоро будет содержать более миллиона треков. Найди свою музыку!
www.yanga.ru
2. Samsung YP-Z5F в металлическом корпусе покажет, что на провалы вкуса вы не жалуетесь.
Z-METAL
www.mp3.samsung.ru
3. Солнцезащитный лосьон Nivea создан для того, чтобы ваша кожа отдыхала комфортно.
4. Когда вы научитесь готовить суши, вам будет жалко кушать это чудо одноразовыми хаси...

www1.mts.ru

на тарифе **«ПЕРВЫЙ»**

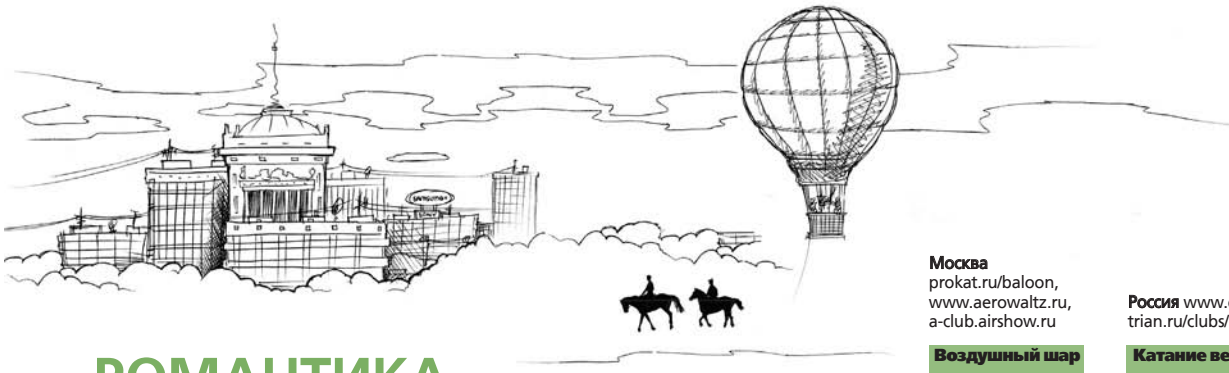
ВСЕ НОМЕРА МТС – ЛЮБИМЫЕ

СКИДКА ОТ 50% НА ЗВОНКИ ВНУТРИ СЕТИ

ПОДКЛЮЧИТЕ УСЛУГУ «НОМЕРА МТС» ПО НОМЕРУ 05906

МТС

На правах рекламы. Скидка действует при подключенной услуге «Номера МТС». Услуга платная. Услуга действует с 30 июня 2006 г. при нахождении абонента в домашней сети. При подключенной услуге скидка распространяется на исходящие вызовы на мобильные телефоны абонентов МТС Вашего региона. Скидка распространяется на стоимость минуты эфирного времени, включая плату за установление соединения. Подробная информация об услуге – на mts.ru и в салонах-магазинах МТС.



РОМАНТИКА

Воздушный шар
Москва
prokat.ru/baloon,
www.aerowaltz.ru,
a-club.airshow.ru

Катание верхом
Россия www.equestrian.ru/clubs/

Гольф
Москва
www.lemeridien-mcc.com,
www.mcgc.ru
Санкт-Петербург
www.golf.spb.ru

Фото: WIRE IMAGES/EASTNEWS

ДЛЯ ТЕХ, ЧЕЙ РАЗУМ НЕ ТРОНУТ ВЛИЯНИЕМ СТЕРЕОТИПОВ, НЕ СЕКРЕТ: РОМАНТИКА – ЭТО ПОДОТРАСЛЬ МАРКЕТИНГА. СЛАЩАВЫЕ РОЗОЧКИ В МЯТОМ ЦЕЛЛОФАНЕ И ДВА СЛИПШИХСЯ СЕРДЦА НА ОТКРЫТКЕ К ДЕЛУ НЕ ОТНОСЯТСЯ. ПОТОМУ ФОРМУЛА УСПЕХА, ГАРАНТИРУЮЩАЯ ОБЕИМ СТОРОНАМ НЕЗЕМНОЕ БЛАЖЕНСТВО, СТАРА, КАК ВОСТОЧНЫЙ БАЗАР: АКТИВНЫЙ ПРОМОУШН, ГРАМОТНОЕ ПОЗИЦИОНИРОВАНИЕ И НЕУЕМНЫЙ КРЕАТИВ.

ВОЗДУШНЫЙ ШАР Все женщины — принцессы, даже если скрывают это в угоду мужским предрассудкам. А у принцесс все должно быть особенным: яблоки — золотыми, туфельки — хрустальными, а пони — розовыми. Воздушный шар — самое что ни на есть особенное средство передвижения. В полете можно угостить спутницу шампанским. Цена — от 12 000 рублей за час.

КАТАНИЕ ВЕРХОМ Все мы немножечко лошади. Попробуйте прокатиться на этих чудных животных, и данное утверждение покажется вам даже немного лестным. Научиться ездить шагом совсем не сложно. К тому же новичкам дают смирных и человеколюбивых животных, а на протяжении всего маршрута вас сопровождает инструктор. Стоимость конной прогулки — от 500 рублей за час.

ГОЛЬФ Когда уже продемонстрированы все цвета маечек Lacoste и процитированы все афоризмы Бернарда Шоу, последний способ подтвердить аристократичность ваших манер — гольф. Поле для гольфа — пересеченная местность, на которой размечено 9 или 18 дорожек-трасс. Игроки имеют наборы клюшек (клэбов), соответствующих различным рельефам и расстояниям. Побеждает игрок, прошедший все лунки при наименьшем количестве ударов. Игра на 18 лунках (без аренды клюшек) стоит 3000 рублей на человека.

Аксессуары

1. Очки Oakley пригодятся везде, где есть хоть лучик солнца, хоть капля воды и хоть легкое дуновение ветерка. Тем более — очки Oakley со встроенным плеером.
2. Необычный букет из фруктов от салона цветов Flower-shop.ru усилит романтическое настроение.
3. Лошади дарят человеку счастье. Если хотите пролонгировать это нестандартное ощущение, купите собственного гнедого-вороного. Или подкову — тоже, говорят, помогает.
4. В верховой езде значение имеет каждая мелочь — например, перчатки.
5. Если вы пока играете в гольф не совсем идеально, советуем не скупиться и приобрести управляемый мячик.

SONY

HANDYCAM

снимайте

смотрите
с объемным
звуком 5.1

делитесь
впечатлениями

DVD Handycam ▶ Снимайте самые ошеломляющие мгновения с объемным звуком 5.1!

Великолепное изображение.
Мощный объемный звук. Небывалая простота использования.
В новом модельном ряду DVD–камер* от Sony захватывающий реализм объемного звучания 5.1 и непревзойденное качество изображения сочетаются с удобством записи непосредственно на диск в DVD–формате.
Записывайте объемный звук прямо во время съемки и все будоражащие звуковые нюансы оживут в полном объеме, как только Вы начнете смотреть записанный DVD со звуком 5.1 на Вашем домашнем кинотеатре. Это все равно, что оказаться на месте съемки вновь!

DOLBY DIGITAL 5.1 CREATOR · RW DVD+ReWritable · DVD R/RW

Модельный ряд DVD–камер 2006 года от Sony:

○ DCR–DVD105* ○ DCR–DVD205* ○ DCR–DVD305 ○ DCR–DVD405 ○ DCR–DVD505

*Модели DCR–DVD105 и DCR–DVD205 не поддерживают функцию записи объемного звука Dolby Digital 5.1

like.no.other™
*Как никто другой

МУЗЕИ

Все о российских музеях
www.museum.ru

Аптекарский огород
Москва
Пр-т Мира, д. 26

Музей экслибриса
Москва
ул. Пушечная, д.7/5

Музей ВВС
Московская обл.
п. Монино;
www.monino.ru

Фото: ИТАР-ТАСС

КТО СКАЗАЛ, ЧТО ХОДИТЬ ПО МУЗЕЯМ СКУЧНО?! ТОТ НАВЕРНЯКА НЕ БЫЛ В МУЗЕЕ СНОВИДЕНИЙ ЗИГМУНДА ФРЕЙДА В САНКТ-ПЕТЕРБУРГЕ, МУЗЕЕ МИРОВОГО ОКЕАНА В КАЛИНИНГРАДЕ ИЛИ МУЗЕЕ ВЕЧНОЙ МЕРЗЛОТЫ В КРАСНОЯРСКОМ КРАЕ. В РОССИИ БОЛЕЕ ТРЕХ ТЫСЯЧ МУЗЕЕВ — СТОИТ ТОЛЬКО ЗАИНТЕРЕСОВАТЬСЯ, И ФЕТРОВЫЕ ТАПОЧКИ СТАНУТ ВАШИМ ALTER EGO.

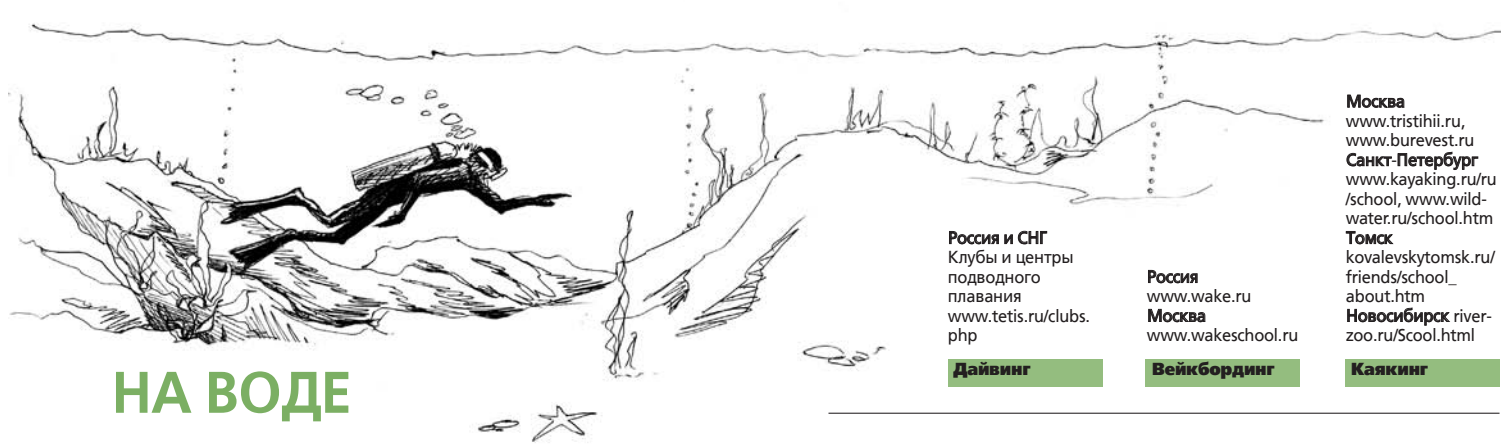
АПТЕКАРСКИЙ ОГОРОД Вечно мятущаяся в электричках душа любителя природы может найти приют в центре Москвы. Аптекарский огород был основан Петром I для выращивания лекарственных растений, а потом приобретен Московским университетом и превращен в Ботанический сад. Отдельным деревьям в парке по 300 лет, среди них лиственница, посаженная все тем же реформатором. Вход бесплатный.

МУЗЕЙ ЭКСЛИБРИСА Экслибрис — это небольшая печать, которой владельцы библиотек помечают свои книги. Экслибрисы бывают гербовые (герб владельца), вензелковые (инициалы владельца) и сюжетные. Есть поистине уникальные экземпляры: сделанные из слоновой кости, нарисованные на рисовых зернышках. Для тех, кто проникся, в мастерской музея могут изготовить экслибрис на заказ. Вход бесплатный.

МУЗЕЙ ВВС Товарищ, верь, взойдет она — возрожденная стратегическая мощь. А пока патриоты и просто технофилы могут отвести душу в Музее ВВС — единственном в стране, основу экспозиции которого составляет реальная авиационная техника: огромный вертолет Ми-12; сверхзвуковой бомбардировщик-ракетоносец Су-100; единственный в России самолет, позволяющий проводить длительные исследования на высотах до 22 км, — М-17, сверхзвуковой пассажирский Ту-144 — советский «конкорд». Билет — 50 рублей.

Аксессуары

1. Если вы не уверены, что привередливая память сохранит все названия авиационных изысков, лучше захватите с собой фотоаппарат, чтобы потом объяснить-таки друзьям, что это была за «штука».
2. С GPS вы не заблудитесь, даже если решите прогуляться от Монино до Музея экслибриса пешком.
3. Главное — всегда быть готовым себя порадовать. Потому, направляясь в Ботанический сад, не забудьте перочинный ножик. Вдруг что приглянется, не с корнем же раритет выдирать.
4. Рюкзак с отделением для охлаждения банок всегда пригодится, если ваш тернистый путь лежит за город.



НА ВОДЕ

Россия и СНГ
Клубы и центры подводного плавания
www.tetis.ru/clubs.php

Дайвинг

Россия
www.wake.ru
Москва
www.wakeschool.ru

Вейкбординг

Москва
www.tristihii.ru,
www.burevest.ru
Санкт-Петербург
www.kayaking.ru/ru/school, www.wild-water.ru/school.htm
Томск
kovalevskytomsk.ru/friends/school_about.htm
Новосибирск river-zoo.ru/Scool.html

Каякинг

Фото: MASTERFILE/EASTNEWS

НЕСМОТРЯ НА РАСПРОСТРАНЕННЫЕ ПРЕДРАССУДКИ, ЛЕТОМ БЫВАЕТ ЖАРКО ДАЖЕ В РОССИИ. ВПРОЧЕМ, НЕ НАСТОЛЬКО ЖАРКО, ЧТОБЫ БЛАЖЕННО ВОЗЛЕЖАТЬ В ЕСТЕСТВЕННОМ ВОДОЕМЕ, ПОТЯГИВАЯ МОХИТО. В НАШИХ ШИРОТАХ СЕНТЕНЦИЯ О ТОМ, ЧТО УМЕНИЕ ВЕРТЕТЬСЯ – ДОСТАТОЧНОЕ И НЕОБХОДИМОЕ УСЛОВИЕ ДЛЯ ЖИЗНИ, НЕ ДОПУСКАЕТ ИСКЛЮЧЕНИЙ.

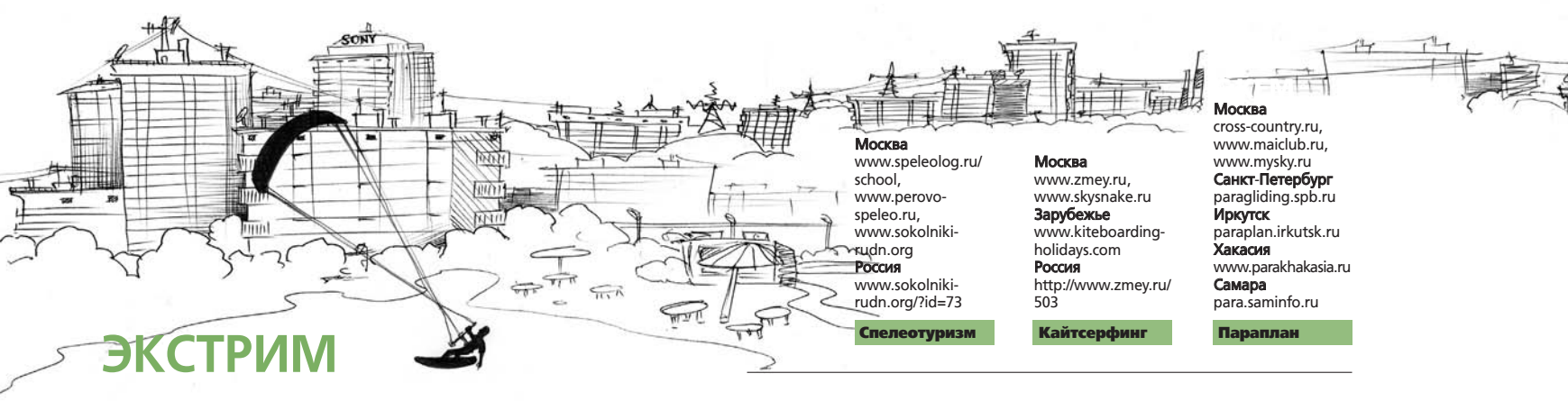
ДАЙВИНГ Можно погрузиться с аквалангом в местном озере. На виды, достойные экспедиций Кусто, вряд ли стоит рассчитывать, но восторг от первого погружения и глобальные ихтиологические открытия гарантируются. Посетив специальные курсы, можете также получить сертификат дайвера. Погружение на открытой воде с инструктором, включая аренду снаряжения, обойдется примерно в 2000 рублей.

КАЯКИНГ Настоятельно рекомендуется поборникам радикального индивидуализма и эскимосской культуры. В экипировку каякера входят лодка с юбкой, которая препятствует попаданию воды в лодку, шлем, жилет и весло. Лучше записаться в школу каякинга. Абонемент на месяц (тренировки 2 раза в неделю) обойдется в 1400 рублей. Знатоки обещают: через год вы поймете, что каяк – неотъемлемая часть вашего тела.

ВЕЙКБОРДИНГ Для тех, кому пацифистские мотивы и лизоблюдские штучки чужды и непонятны, существует вейкбординг – аналог сноуборда на воде. Райдер встегивается в крепления, недрогнувшей рукой берет фал (веревка с рукояткой) и ждет, пока тронется катер, с коего ему этот фал и кинули. Волна, которую катер оставляет за собой, используется в качестве трамплина. Средняя скорость передвижения – 30-40 км/ч. Не каждый катер подойдет для занятий вейком, а потому спорт не самый дешевый: 600 рублей за 10 минут катания.

Аксессуары

1. Вейкборд собственной персоной.
2. Водная маска Oakley защитит от назойливых брызг и обеспечит полный обзор.
3. Гидрокостюм пригодится, каким бы видом водного спорта вы ни занимались, — и не замерзните, и никакой водной заразы не подхватите.
4. К сожалению, люди порой вспоминают о необходимости спасательного жилета только в экстренных случаях, когда требуется уже не жилет, а служба спасения на водах.
5. Походный рюкзак Deuter Aircontact 40+10 SL для каякинга настоящая находка — анатомические лямки, многочисленные продуманные кармашки и плотная ткань.



ЭКСТРИМ

Спелеотуризм

Москва
www.speleolog.ru/school,
www.perovo-speleo.ru,
www.sokolniki-rudn.org
Россия
www.sokolniki-rudn.org/?id=73

Кайтсерфинг

Москва
www.zmey.ru,
www.skysnake.ru
Зарубежье
www.kiteboarding-holidays.com
Россия
http://www.zmey.ru/503

Параплан

Москва
cross-country.ru,
www.maiclub.ru,
www.mysky.ru
Санкт-Петербург
paragliding.spb.ru
Иркутск
paraplan.irkutsk.ru
Хакасия
www.parakhakasia.ru
Самара
para.saminfo.ru

СРЕДИ ВСЕХ «ИЗМОВ» И «МАНИЙ», КОТОРЫМИ ОБРОСЛО ЧЕЛОВЕЧЕСТВО, ТОЛЬКО ОДНА ФОРМА ЗАВИСИМОСТИ НЕ СТАЛА ОБЩЕСТВЕННО ПОРИЦАЕМОЙ – АДРЕНАЛИНОМАНИЯ. ДОЗА АДРЕНАЛИНА В КОМПЛЕКСЕ С ЭНДОРФИНАМИ ВЫЗЫВАЕТ, С ОДНОЙ СТОРОНЫ, ИЗДРЕВЛЕ ТАБУИРОВАННЫЙ ВО ВСЕХ ЕГО ПРОЯВЛЕНИЯХ «КАЙФ», А С ДРУГОЙ – НЕ ВРЕДИТ ЗДОРОВЬЮ. СЛОВОМ, ЗАПРЕТНЫЙ ПЛОД, ОБОГАЩЕННЫЙ ВИТАМИНАМИ.

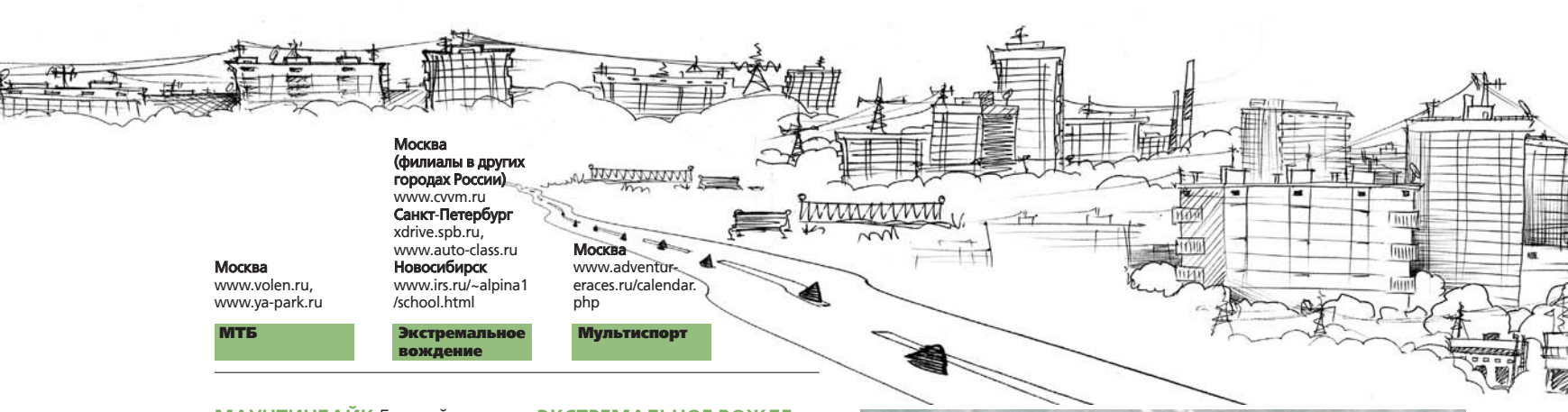
СПЕЛЕОТУРИЗМ Спелеотуризм – это путешествия по пещерам. Если Платон призывал всех выползти из пещеры, ассоциировавшейся у него с неведением, то современная цивилизация, напротив, предлагает туда спуститься. Но опять-таки – исключительно в общеобразовательных целях. Цены на один поход с инструктором начинаются от $50 за подмосковную «пещеру».

КАЙТСЕРФИНГ Человек всегда стремился разделять и властвовать. На худой конец, объединять и доминировать. Объединить и подчинить две стихии – не лейтмотив фильма «Шестой элемент», а краткое описание кайтсерфинга. Этот вид спорта крайне демократичен: требуется лишь вода, воздушный змей (кайт) и доска. Занятия с инструктором (включая аренду снаряжения) стоят 1000 руб/час.

ПАРАПЛАН Люди не летают, как птицы, по вполне объективным причинам: люди, в отличие от птиц, существа с многоплановым мышлением. А потому они изобрели параплан – летательный аппарат сверхлегкой авиации, который позволяет планировать в воздухе часами и пролетать десятки километров без особой нервотрепки и прочих мозолей величиной с кулак. Возможно, вы не сразу покорите премудрости аэробатики (акробатики в воздухе), но прыжок тандемом с инструктором за 600 рублей по силам каждому.

Аксессуары

1. Не забудьте камеру Sony DCR-DVD505E, чтобы запечатлеть все свои экстремальные подвиги. Датчик ClearVid CMOS обеспечит отменное качество изображения, а ЖКД 3,5" — удобный просмотр. В камере есть возможность 5.1-канальной записи, то есть все вопли встречающих вас на земле фанатов войдут в историю в формате «звук вокруг».
2. Чтобы знать, на какую высоту вы поднялись, рекомендуем использовать часы с высотомером Tissot.
3. Если пещеры наводят вас на мысли об антисанитарии, а не о прогрессивном отдыхе, купите себе часть земных недр в уже обработанном варианте.
4. Доска для кайта.



МТБ

Москва
www.volen.ru,
www.ya-park.ru

Экстремальное вождение

Москва (филиалы в других городах России)
www.cvvm.ru
Санкт-Петербург
xdrive.spb.ru,
www.auto-class.ru
Новосибирск
www.irs.ru/~alpina1/school.html

Мультиспорт

Москва
www.adventur-eraces.ru/calendar.php

МАУНТИНБАЙК Горный велосипед — универсальный вид спорта, который от этого, впрочем, не становится менее увлекательным. Хотите — катайтесь по паркам, хотите — прыгайте на специальных велосипедных трассах. Более того, можно просто колесить по городу — городское катание («стрит») становится с каждым годом все популярнее. В некоторых подмосковных парках есть специально подготовленные трассы для байкер-кросса с бугельными подъемниками. Начинающие могут взять горный велосипед напрокат (250 руб/час) или воспользоваться услугами инструктора (от 200 рублей за час).

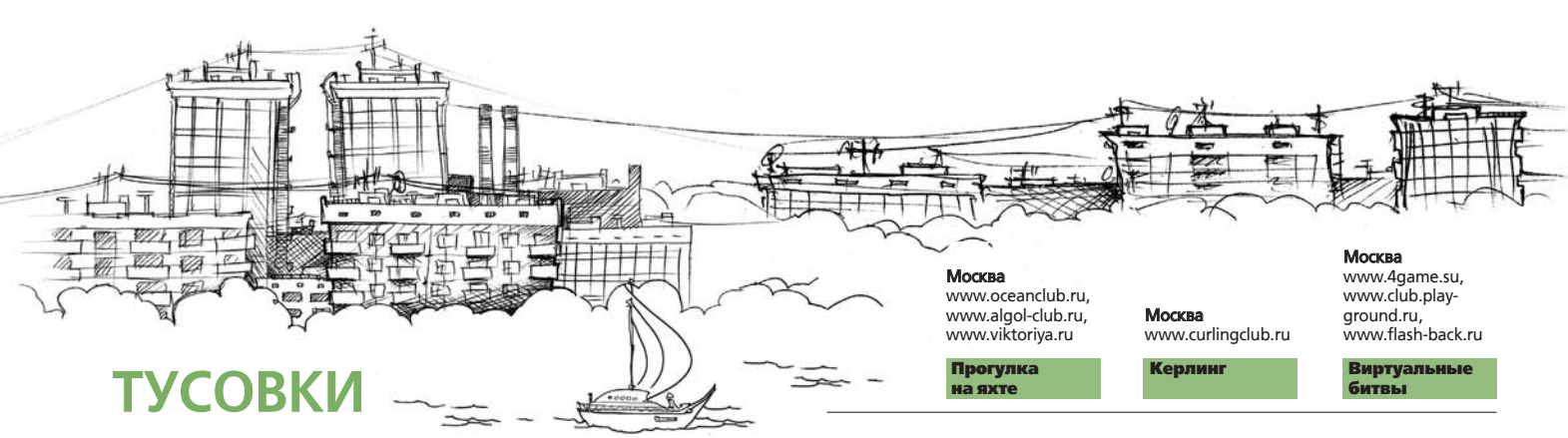
ЭКСТРЕМАЛЬНОЕ ВОЖДЕНИЕ Быстрой езды не любят только те, кто не умеет ездить быстро. Чтобы не попасть в прискорбную категорию «чайников» поневоле, можно посетить школу экстремального вождения. Там моделируются реальные аварийные ситуации: невозможность объехать препятствие или остановить автомобиль, уход от столкновения, торможение на скользкой дороге, прохождение поворотов в управляемом заносе, «полицейский» разворот. Начальную контраварийную подготовку можно пройти на специальных тренажерах. Стоимость базового обучения в группе составит около 11 000 рублей.

Фото: Татьяна Чехова

Аксессуары

1. iPod, совместимый с большинством автомобильных аудиосистем, позволяет слушать любимую музыку в любых, даже самых экстремальных ситуациях.
2. Портативный видеомагнитофон предназначен для активных людей, которые любят быть в курсе всего происходящего.
3. Тренажер Powerball не только тренирует предплечья, но и успокаивает нервную систему. Просто зажмите его в кулаке и вращайте кистью.
4. Велокомпьютер — незаменимая вещь. Вы будете знать не только свою среднюю и максимальную скорость, но даже количество потраченных калорий, общий километраж и количество оборотов педалей.

МУЛЬТИСПОРТ Можно попробовать силы в мультиспортивной приключенческой гонке. Это сочетание спортивного ориентирования (перемещение с помощью карты и компаса через контрольные пункты) с различными видами спорта – бег, велосипед, плавание, скалолазание, рафтинг, верховая езда, ролики. Гонки могут проводиться по маркированному маршруту со сменой экипировки на каждом этапе, а могут представлять собой автономное преодоление трассы со всей экипировкой. Стартовый взнос — от 200 рублей.



ТУСОВКИ

Прогулка на яхте
Москва
www.oceanclub.ru,
www.algol-club.ru,
www.viktoriya.ru

Керлинг
Москва
www.curlingclub.ru

Виртуальные битвы
Москва
www.4game.su,
www.club.play-ground.ru,
www.flash-back.ru

Фото: GETTY/PHOTONICA/FOTO S.A.

ЛЕТО – САМОЕ ВРЕМЯ ДЛЯ ТУСОВОК С ДРУЗЬЯМИ. НО В НОЧНЫХ КЛУБАХ ДУШНО, ШАШЛЫКИ НАДОЕЛИ, НА ДАЧЕ – РОДИТЕЛИ, В ГОСТЯХ СКУЧНО. ОТЧАИВАТЬСЯ РАНО: НЕМНОГО ФАНТАЗИИ, НЕСТАНДАРТНЫЙ ПОДХОД К ВЫБОРУ МЕСТА ВСТРЕЧИ, И ДРУЗЬЯ В ПОЛНОМ ВОСТОРГЕ.

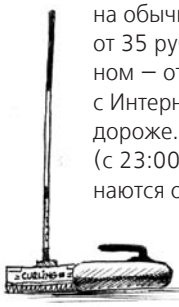
ПРОГУЛКА НА ЯХТЕ Можно натолкать друзей на яхту и устроить вечеринку под парусами. Шампанское льется рекой, на палубе красивые загорелые полуголые фигуры. Красиво, незаезженно, а главное – весело и не жарко. Если захотелось освежиться, капитан бросит якорь в живописной бухте, и всей компанией можно попрыгать в воду. Аренда 17-метровой парусной яхты на компанию из 10 человек стоит от 2000 рублей за час.

КЕРЛИНГ Если столбик термометра поднялся выше +30 и опускаться не собирается, пригласите друзей сыграть в керлинг. Цель этой командной игры на льду – попасть пущенной по льду специальной битой из камня в мишень. Команда состоит из 4 человек. Каждый по очереди пускает снаряд, а остальные натирают лед щетками по ходу движения снаряда, чтобы скорректировать дальность скольжения и траекторию. Аренда дорожки – от 2500 рублей.

ВИРТУАЛЬНЫЕ БИТВЫ
Отправившись шумной компанией в интернет-кафе, вы сможете и поохотиться друг на друга в Counter-Strike, и коллективно пройти какую-нибудь онлайн-игру (например, World of Warcraft), и померяться силами в более мирные игры – такие как FIFA Soccer 2006, Need For Speed Most Wanted. Цена за час на обычном компьютере от 35 рублей, на наворочен-ном – от 60 рублей. Машина с Интернетом обойдется подороже. Для ночных жителей (с 23:00 до 9:00) цены начинаются от 120 рублей.

Аксессуары

1. Развлекательная система PSP — это мир развлечений, который всегда с тобой. Крутые трехмерные игры, фильмы, музыка, фотографии, выход в интернет – малышка от Sony умеет все! Остается лишь один вопрос: классический черный или элегантный белый?
2. Дабы сберечь свое доброе здравие, не забудьте дома ветровку 686 Ard Transition Windbreaker.
3. Специальных наколенников для керлинга еще нет, используйте роликовые.
4. Настоящего морского (или речного) волка невозможно вообразить без трубки.
5. Каждый уважающий себя геймер ходит в клуб с собственной мышкой.

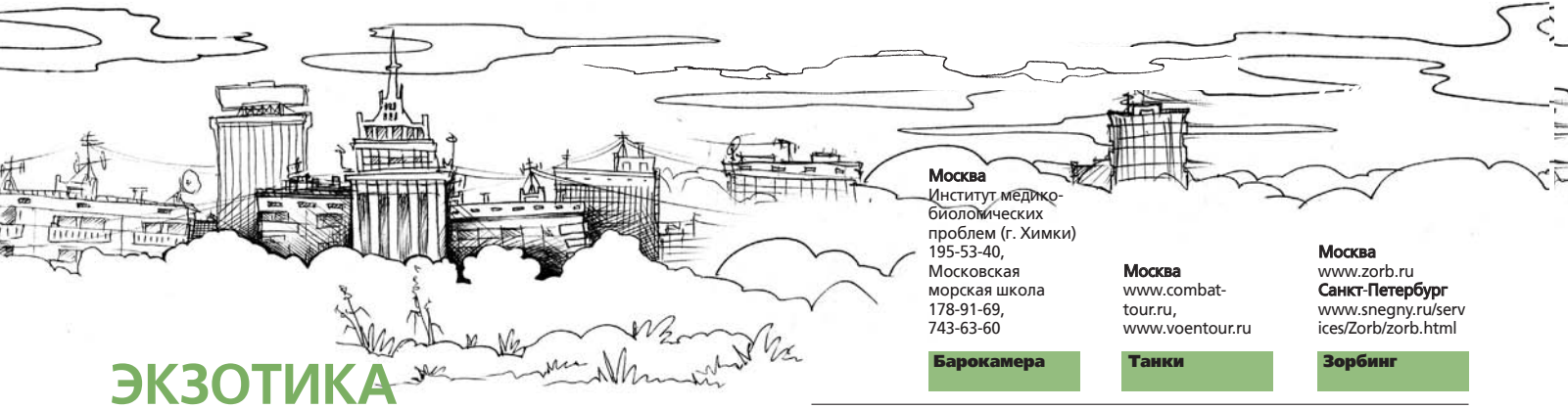


SONY
COMPUTER ENTERTAINMENT

БЕЛАЯ ИЛИ ЧЕРНАЯ?

www.playstation.ru www.mypsp.ru

PSP



ЭКЗОТИКА

Барокамера
Москва
Институт медико-биологических проблем (г. Химки)
195-53-40,
Московская морская школа
178-91-69,
743-63-60

Танки
Москва
www.combat-tour.ru,
www.voentour.ru

Зорбинг
Москва
www.zorb.ru
Санкт-Петербург
www.snegny.ru/services/Zorb/zorb.html

Фото: Евгений Ветлугин

СОТВОРИТЬ НЕЧТО ЗАПОМИНАЮЩЕЕСЯ, НО НЕ ПРОТИВОРЕЧАЩЕЕ НОРМАМ МОРАЛИ, ТЯЖЕЛО. БЫТУЕТ МНЕНИЕ, ЧТО ВСЕ НЕСТАНДАРТНОЕ ГНЕЗДИТСЯ ЛИБО НАД ОБЛАКАМИ РАЗУМНОЙ ЦЕНЫ, ЛИБО НА ГЛУБИНЕ 4000 МЕТРОВ, ЛИБО В ДРУГОМ МЕСТЕ, ДО КОЕГО ОТ ЧЕЛОВЕЧЕСКОГО ЖИЛЬЯ ПЯТЬ ЧАСОВ ПОЛЗКОМ ПО ДЖУНГЛЯМ, БОЛОТАМ И ОТВЕСНЫМ СКАЛАМ. ОТНЮДЬ — НАЙТИ НЕОБЫЧНЫЕ РАЗВЛЕЧЕНИЯ МОЖНО И В РОДНОМ ГОРОДЕ.

БАРОКАМЕРА Группу из двух-трех человек помещают в герметически закрывающуюся камеру и искусственно создают в ней давление, как на глубине 80 или 100 м. Дело в том, что на глубине у человека возникает состояние, которое называют азотным опьянением. Чем глубже, тем больше голоса погружаемых напоминают писк Пятачка. Погружение на 80 м стоит 2500 рублей.

ТАНКИ Если мировые войны и глобальные столкновения биполярного мира отгрохотали без вашего непосредственного участия, прокатитесь хотя бы на БТР или танке тех славных времен. Ваш экипаж сможет испытать танки эпохи ВОВ (советские Т-34 и ИС-3, немецкий Pz-IV), периода Вьетнама (Т-54) и Афгана (Т-72), современный Т-84. Полчаса на БТР — 14 000 рублей, а на Т-34 — 22 000 рублей.

ЗОРБИНГ Способ передвижения Джеки Чана в фильме «Доспехи Бога: Операция "Кондор"». Человека помещают в прозрачный надувной шар и спускают с горы.
За каждые десять метров пути (обычная трасса — 300 метров) зорб делает один полный оборот. Так что вращение зорбонавта, который «бежит» внутри капсулы, получается в итоге вполне степенным и абсолютно безопасным. Один спуск стоит 300 рублей. К слову, в Москве в скором времени собираются построить внесезонную трассу для зорба.

Аксессуары

1. Стильный корпус флэш-плееров BBK серии X21 изготовлен из алюминия и имеет необычный дизайн. Современный аудиопроцессор Philips и система цифровой обработки звука Life Vibes позволяют получить качественный и насыщенный звук. Кроме того, плееры серии имеют FM-тюнер с памятью на 20 станций, функцию диктофона и эквалайзер с возможностью ручной настройки. Серия X21 будет выпускаться в двух вариантах — 12 Мб и 1 Гб. В комплекте высококачественные наушники Sennheiser.
BBK X21
www.bbk.ru
2. Триумфально осматривать окрестности сподручнее при помощи качественной оптики.

3. Рекомендуем соблюдать дресс-код: в тельняшке с начесом вы почувствуете себя настоящим подводником.
4. Зорб.

До **55%** НА ВСЁ ЛЕТНЕЕ

Пилим цены!

КАНТ

www.kant.ru

ЛЕГКО ВЫБРАТЬ СВОЕ!

Москва
м.Нагорная, Электролитный проезд, вл. 7б тел. 317-61-01
м.Полежаевская, ул. Куусинена, д.9, тел. 943-11-55
Санкт-Петербург
м.Академическая, Гражданский пр-т, д.23, тел. 535-33-91
м.Ломоносовская, ул. Ивановская, д.7
тел. 560-06-60, 560-61-00
Самара
Проспект Ленина, д.1, тел. 338-17-55

Возможны изменения цен при изменении курса евро

СИТИ-БРЕЙК

/АВТОРЫ ИДЕИ
ПАША РОМАНОВСКИЙ
(ROMANOVSKI@GAMELAND.RU)
ДАВИД ШОСТАК
(SHOSTAK@GAMELAND.RU)

/РЕДАКЦИЯ
>ГЛАВНЫЙ РЕДАКТОР
ДМИТРИЙ РЫВКИН (RYVKIN@GAMELAND.RU)
>ВЫПУСКАЮЩИЙ РЕДАКТОР
МАРИЯ СОБОЛЕВА (SOBOLEVA.M@GAMELAND.RU)

/ДИЗАЙН
>АРТ-ДИРЕКТОР
ЕЛЕНА ТИХОНОВА (TIHONOVA@GAMELAND.RU)
>ХУДОЖНИК
ОЛЕГ БАСКОВ

/ОТДЕЛ РЕКЛАМЫ
>РУКОВОДИТЕЛЬ ОТДЕЛА РЕКЛАМЫ
ИГОРЬ ПИСКУНОВ (IGOR@GAMELAND.RU)

/КОРПОРАТИВНЫЙ ОТДЕЛ
ЛИДИЯ СТРЕКНЕВА (STREKNEVA@GAMELAND.RU)

/ИЗДАТЕЛЬСТВО
>ИЗДАТЕЛЬ
БОРИС СМИРНОВ (BORISSMIRNOV@GAMELAND.RU)
>ГЕНЕРАЛЬНЫЙ ДИРЕКТОР
ДМИТРИЙ АГАРУНОВ (DMITRI@GAMELAND.RU)
>ДИРЕКТОР ПО РАЗВИТИЮ
ПАША РОМАНОВСКИЙ (ROMANOVSKI@GAMELAND.RU)
>УПРАВЛЯЮЩИЙ ДИРЕКТОР
ДАВИД ШОСТАК (SHOSTAK@GAMELAND.RU)
/АДРЕС РЕДАКЦИИ
МОСКВА, УЛ. ТИМУРА ФРУНЗЕ, Д. 11, СТР. 44-45,
ТЕЛ.:(495)935-70-34, ФАКС:(495)780-88-24

(game)land