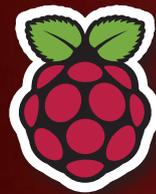


НА ВАШЕМ БЕСПЛАТНОМ DVD
Ubuntu, Kubuntu, Xubuntu
Manjaro, книги, подкасты и более того!

LINUX FORMAT

Главное в мире Linux



Raspberry Pi

» Изучим программирование на Scratch с нуля! с. 66

Июль 2013 № 7 (172)

Что утратил Ubuntu?

Unity » Раскол разработчиков
Тревога сообщества » MIR

Требуем объяснений
от любимого во всем мире
дистрибутива



ПЛЮС!
X.org: Конец истории
с. 54

Erlang Задач меньше не стало
KVM и RPyC Как построить кластер
GIMP Я гляжу на фотокарточку...



Kerbal Space Program

» Колонизируйте Марс, сидя в уютном кресле — благодаря Linux и Steam с. 20



Лисьи телефоны

« Одна из целей OS — приложения должны быть кросс-платформенными »

Джонас Сикинг дело говорит с. 38

Также в номере...

По касательной

К этим дистрибутивам надо притронуться с. 24



Музыка с Ardour

Введение в классный комплект для производства аудио с. 76



Trisquel GNU/Linux внутри

Там, где свобода — во главе угла с. 42



Защита данных
Копии — легко

» Продублируйте файлы, не дожидаясь катастрофы

Фильтрация процессов
Хватка родителя

» Уберегите детишек от худшего в Интернете

Генеалогия
Дорогие покойники

» Отыщите своего Ричарда III с помощью Gramps

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

Linux center
www.linuxcenter.ru

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

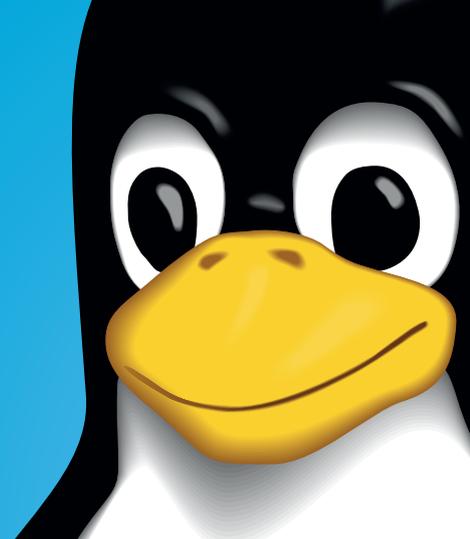
Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center

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 дружелюбных 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

Добро пожаловать в робототехнику!

ScratchDuino

Электронный комплекс на основе свободного аппаратного обеспечения для школ и вузов

ScratchDuino.Лаборатория

Плата расширения с датчиками ввода-вывода — как установленными на плате, так и подключаемыми. Предназначена для программирования взаимодействия компьютера с внешними устройствами.

Полная интеграция («из коробки») со средой программирования Scratch, предназначенной для детей.

ДАТЧИК ЗВУКА*

ДАТЧИК ОСВЕЩЕННОСТИ*

КРЕСТОВИНА КНОПОК

ЦВЕТНЫЕ СВЕТОДИОДЫ

ДИНАМИК ДЛЯ ВЫВОДА ЗВУКА



ПЕРЕМЕННЫЙ РЕЗИСТОР (ПОЛЗУНОК)*

КАРТРИДЖ ARDUINO

СВЕТОДИОДЫ, УПРАВЛЯЕМЫЕ ЧЕРЕЗ РАЗРЯДНУЮ МАТРИЦУ

РАЗЪЕМЫ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ

КНОПКА*

* РАБОТАЮТ В РЕЖИМЕ ЭМУЛЯЦИИ RISCBOARD, НЕ ТРЕБУЯ ДОПОЛНИТЕЛЬНОГО ПРОГРАММИРОВАНИЯ.

ЦЕНА **5000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

ScratchDuino.Робоплатформа

Внешний робот-исполнитель, управляемый из среды программирования Scratch. Не требует навыков программирования на языках высокого уровня и может применяться для обучения, начиная с младших классов.

ScratchDuino.Робоплатформа

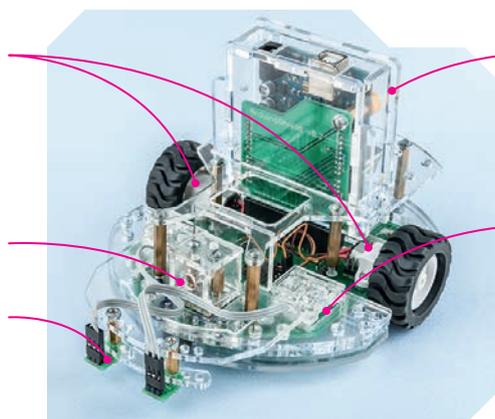
обеспечивает возможность:

- подключения внешних датчиков без механических соединений и пайки;
- расширения через последовательную шину RoboBus;
- установки деталей Lego Technics.

МОТОР-РЕДУКТОРЫ

ФОТОДАТЧИК

ДАТЧИК ЛИНИИ



КАРТРИДЖ ARDUINO

МАГНИТНЫЙ ЦИФРО-АНАЛОГОВЫЙ РАЗЪЕМ ДЛЯ ПОДКЛЮЧЕНИЯ ВНЕШНИХ ДАТЧИКОВ БЕЗ МЕХАНИЧЕСКИХ СОЕДИНЕНИЙ И ПАЙКИ*

* ВСЕГО НА РОБОПЛАТФОРМЕ 5 РАЗЪЕМОВ. ВОЗМОЖНО ПОДКЛЮЧЕНИЕ ДАТЧИКОВ КАСАНИЙ, ИК-ДАТЧИКОВ РАССТОЯНИЙ/ПРЕПЯТСТВИЙ И ДР.

ЦЕНА **10 000 руб.**

ГАРАНТИЯ 3 ГОДА, ВКЛЮЧАЯ ТЕХНИЧЕСКУЮ И МЕТОДИЧЕСКУЮ ПОДДЕРЖКУ

Назначение ScratchDuino

- Изучение взаимодействия компьютера с внешней средой
- Изучение изменения параметров внешней среды
- Изучение процессов передачи информации и принципов ее построения
- Изучение внешних устройств управления
- Моделирование устройств

Среда разработки

Lazarus (язык Pascal) ■ Scratch ■ Arduino IDE

ScratchDuino адаптирован для учебных заведений. Поставляется с комплектами учебно-методических материалов.

Продукт разработан при финансовой поддержке Фонда содействия развитию малых форм предприятий в научно-технической сфере

WWW.SCRATCHDUINO.RU

(812) 309-0686 (Санкт-Петербург) ■ (499) 271-4954 (Москва)

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

В данном номере мы рассмотрели смену направления Ubuntu. Если бы вы создали свой ремикс Ubuntu, как бы вы его назвали?



Гэри Уокер

Не понял, почему вы меня об этом спрашиваете — я же месяц назад уволился.



Эндрю Грегори

YAUbuntu, потому что в номере про Ubuntu можно было бы задать вопрос и получить.



Эфраин Эрнандес-Мендоса

IceManbuntu: должен же быть дистрибутив про мой любимый «Лучший стрелок».



Бен Эверард

Извините, не могу ответить: я сейчас пересекаю Шотландию на деревянном велосипеде.



Маянка Шарма

Есть у меня диск Ubuntu 8.10, и я им премного доволен, спасибо. Эх, денечки были...



Джонатан Робертс

А можно, я просто скопирую ответ с предыдущего номера, как Маянка?



Майк Сондерс

Так и быть, про MikeOS я на сей раз упоминаю не стану и выберу что-нибудь другое. Например, M...



Валентин Синицын

Смена век? Тогда — Antibuntu со сложным интерфейсом в желтых тонах... хотя это уже «Роботрон» какой-то получается.



Ник Вейч

Попробовал бы кто-нибудь задавать подобные вопросы в мое время...



Сюзан Линтон

Статьи в этом номере сожгла мосты. Лучше перейду на Fuduntu. Погодите...



Шашанк Шарма

Shubuntu.



Нейл Боткич

Бен отлично поработал над DVD этого месяца, так что я выбираю LXF Privacy Remix оттуда.



Проблема выбора

» Народная мудрость гласит, что наши недостатки суть продолжение наших достоинств. Статья о недоработках Ubuntu наводит на мысль, что этот принцип распространяется и на дистрибутивы. Основные претензии к этому и многим другим дистрибутивам можно свести к одному вопросу: зачем?

Зачем менять привычный *init* на *systemd*? Зачем менять проверенные годами «иксы» на какой-то новомодный Wayland или непонятный Mir? Зачем переходить с удобного «2-го Гнома» на Unity?.. и так далее по списку What's new. Иногда возникает мысль, что прогресс технологии пошел не в ту сторону, и изменения вносятся исключительно из желания чем-то отличаться от других. Каждая подобная инновация вызывает яростные споры и бурные обсуждения.

Охота к радикальным переменам характерна и для проприетарных систем. Лучший пример — интерфейс Metro в Windows 8. В этом случае споров нет, только глухое недовольство — а нам теперь с этим жить...

Зато в мире свободного ПО всегда можно найти альтернативу. И не одну. Ничто хорошее не исчезает бесследно, и каждый может работать в привычной среде. Не нравится Ubuntu с Unity — возьмите Mint с MATE... Единственная проблема — богатство выбора. И в этом сила.

Кирилл Степанов

Главный редактор

» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru

Подписка и предыдущие номера: subscribe@linuxformat.ru

Техническая поддержка: answers@linuxformat.ru

Общие вопросы: info@linuxformat.ru

Проблемы с дисками: disks@linuxformat.ru

Вопросы распространения: sales@linuxformat.ru

Сайт: www.linuxformat.ru, группа «ВКонтакте»: vk.com/linuxform

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

» Телефон редакции: (812) 309-0686. Дополнительная информация на с. 112

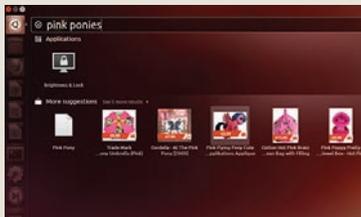
Содержание

Не столько операционная система, сколько образ жизни.

Обзоры

Ubuntu 13.04 16

Последний релиз популярного дистрибутива исходит зрелость и удобство в использовании.



Ищите розовых пони через усовершенствованный портал Dash.

Ciseco XinoRF 18

Микроконтроллерная плата, совместимая с Arduino, для физических компьютерных проектов.

Manjaro Linux 19

Дистрибутив с возобновляемыми релизами на базе Arch сулит и мощь, и дружелюбие к пользователю.

Kerbal Space Program 20

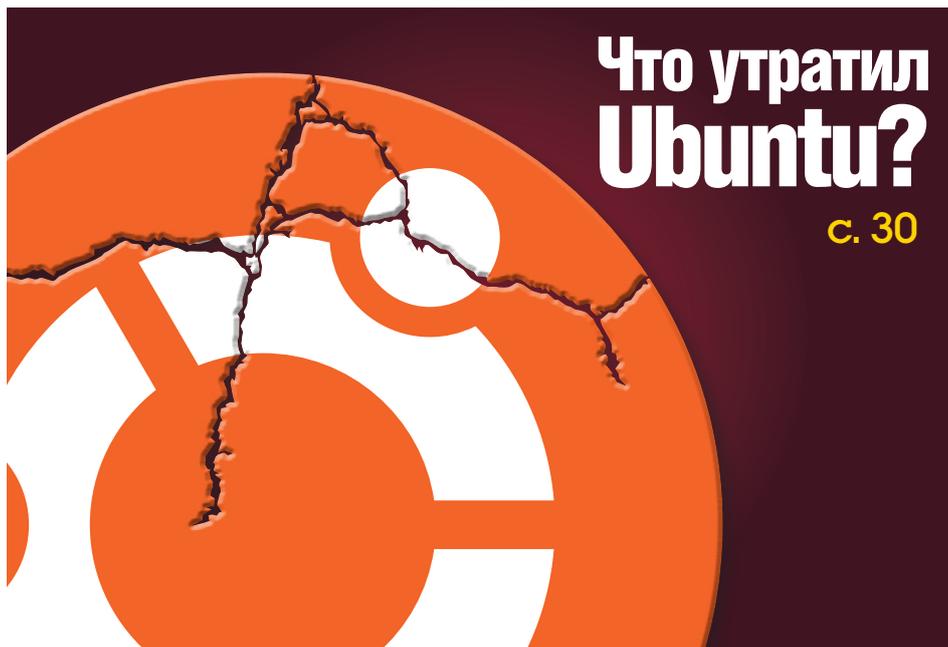
Проектируйте и пилотируйте космолеты в игре — песочнице ракетной техники.



Пускай персонажи и китчевые, но имитатор космоса вполне серьезный.

OPPO Find 5 21

Смартфон из Поднебесной, чтобы победить Apple и Samsung, продает высокое качество по низкой цене.



Сравнение: Дистрибутивы для планшет- ников x86 с. 24



Люди говорят

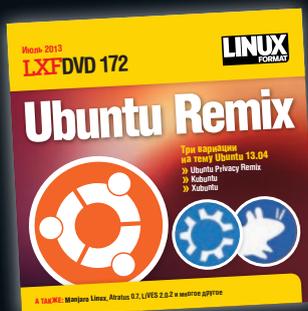


“Это эксперимент — посмотрим, что работает, что понравится пользователям”

Джонас Сикинг — об открытых мобильных платформах с. 38

TOP10
Награды нашли
своих героев
с. 13

На вашем бесплатном DVD



Ubuntu 13.04

» С улучшенной приватностью

Manjaro

» Передовой и дружелюбный

ПЛЮС: HotPicks и коды учебников с. 106

Ищите в этом номере



Trisquel изнутри 42

Соратник FSF в борьбе за свободу ПО.

Властелин былого 46

Исследуйте свою родословную с Gramps.

Мини-компьютеры 50

Их развелось великое множество.



Пропустили номер?

Узнайте на с. 108, как получить его прямо сейчас!



Учебники

Raspberry Pi
На языке малышей 66
 Запрограммируем на языке Scratch игру в кошки-мышки.

Программирование
Основы Bash 70
 Разберемся, как использовать оболочку Bash и язык написания скриптов.

Резервное копирование
Duplicity 74
 Довольно терять файлы! Продублировать данные вовсе не трудно.

Создание музыки
Ardour 76
 Освоим превосходную рабочую станцию для цифровых композиторов.

Родительский контроль
Детки без беспредела 80
 Простые скрипты ограничат пребывание ваших отпрысков онлайн и за компьютером.

Программирование
Erlang 84
 В очередной раз оттачиваем многозадачные версии функций.

Параллельные вычисления
Кластеры 88
 Оказывается, вычислительный кластер можно сделать и на дому, было бы желание.

Обработка изображений
GIMP 92
 Пожелтевшие фото с бабушкиной свадьбы будут выглядеть в компьютере как новенькие.

Постоянные рубрики

Новости 6
 ROSA выпускает дистрибутив и курирует Google Summer of Code, Берлин отвергает СПО, а Мюнхен продвигает, Intel задумалась о мобильности, а Canonical исправил ошибку № 1.

Новости Android 22
 В Москве прошла выставка-конгресс MATE-2013, а мобильные приложения тоскуют по стандартам.

Сравнение 24
 Душа просит Linux на планшете? Ну так ознакомьтесь для начала с нашим исследованием.

Интервью LXF 38
Джонас Сикинг наслаждается атмосферой инженерного подхода к разработкам.

Что за штука... Tizen 58
 Открытая платформа, которой ждут не дожидаться мобильные устройства.

Рубрика сисадмина 60
 Доктор увлекся тяжелым металлом, предоставляемым как сервис, и продолжает настройку сервера Tomcat.

Ответы 96
ВАШИ ПРОБЛЕМЫ РЕШЕНЫ!
Нейл Ботвик — про собой с установкой обоев, клонирование готовой системы, печать файлов man/info, проблемы с разбиением диска и прочее.

Hotpicks 100
 Отведаете горяченького: лучшие в мире новинки свободного ПО. Atratus, Catfish, Choria, Colorful, ETM, G4L, Gitg, LIVES, Plgen, Siege, Sxiv

Диск Linux Format 106
 Содержимое двустороннего DVD этого месяца.

Пропустили номер? ... 108
 Коллекция неполна? Если на вашей полке с журналами образовалась ниша размером с LXF, вы знаете, куда обратиться.

Через месяц 112
 Крылатая колесница Времени стремится вперед — что же принесет она LXF173?



ГЛАВНОЕ Свежая ROSA » Германия против патентов » Новые версии Tizen
» Прогресс Ubuntu Touch

НАСЛЕДНИК MANDRIVA

Вышел дистрибутив ROSA Desktop



» Рубрику готовил
АРТЕМ ЗОРИН

А эксперты «РОСА» и в 2013 году будут курировать проекты Google Summer of Code.

27 мая был опубликован список проектов, которые будут выполняться в рамках программы Google Summer of Code™ 2013. Это ежегодная программа компании Google, в рамках которой студентам и аспирантам выделяются гранты для выполнения определенных работ для открытых проектов. В 2013 году такую материальную поддержку получают 1192 студента, привлеченные к задачам для 177 открытых проектов.

В двух проектах под патронажем The Linux Foundation менторами станут эксперты компании «РОСА» — Владимир Рубанов, Денис Силаков (старший архитектор) и Евгений Шатохин. Под руководством В. Рубанова и Д. Силакова будет реализована поддержка мультиверсионности в ряде тестовых наборов, применяемых для проверки соответствия дистрибутивов требованиям стандарта Linux Standard Base (LSB). Е. Шатохин будет курировать проект по увеличению производительности инструментария KernelStrider, используемого для обнаружения нарушений синхронизации доступа к данным из разных потоков в модулях ядра Linux. KernelStrider является частью KEDR — системы динамического анализа модулей ядра Linux, используемой, в частности, для автоматизированной проверки драйверов.

«Тестовые наборы LSB и система KEDR активно используются при тестировании дистрибутивов «РОСЫ». Мы надеемся, что улучшение этих инструментов позволит нам еще больше повысить качество наших продуктов», отмечает Силаков.

Эксперты «РОСЫ» не первый год выступают в роли менторов. Являясь членами рабочей группы Linux Standard Base, Рубанов и Силаков с 2009 года курируют

проекты по LSB, а также разработку инструмента Linux Application Checker. Шатохин — основной разработчик системы KEDR, и в 2011 году курировал проект по добавлению в KEDR поддержки перехвата обратных вызовов в модулях ядра Linux.

ЗАО «РОСА» является российским центром разработки широкого спектра решений на основе свободного ПО. В линейке продуктов компании — настольные, серверные, мобильные и встраиваемые ОС и решения с оригинальным дружелюбным интерфейсом, инфраструктурные сервисы и прикладное ПО. Решения компании отмечены различными наградами — например, BEST SOFT 2012 журнала PC Magazine/RE.

Еще одна приятная новость для поклонников «наследницы» Mandriva: «РОСА» выпустила свежий релиз своего дистрибутива для Linux-сообщества — ROSA Desktop Fresh R1.

ROSA Desktop Fresh R1 — новое имя для дистрибутивов на базе платформы ROSA Fresh. Линейка «R» предназначена для технически грамотных пользователей и энтузиастов, ценящих функциональную насыщенность и новизну компонентов дистрибутива без существенного падения качества. Данное семейство дистрибутивов развивается как силами компании, так и силами сообщества.

Среди ключевых нововведений:

» Поддержка Microsoft Hyper-V позволяет размещать ROSA в публичных облачных дата-центрах Microsoft Azure и частных облаках для организации виртуальных рабочих столов. В каталоге VM Depot скоро будут размещены готовые образы системы для разворачивания ROSA в Azure «из коробки».



» Основной упор в ROSA Desktop Fresh R1 сделан на опытных пользователей.

» Поддержка Steam дает возможность подключения к игровому сервису и получения доступа ко множеству игр под Linux.

» Внесены изменения в окружение рабочего стола: улучшено сглаживание шрифтов, благодаря новому растеризатору от Adobe и Google, а также переработана фирменная графическая тема ROSA. Обновлены популярные приложения LibreOffice, Firefox, Thunderbird и Chromium.

» В системной части обновлено ядро системы до версии 3.8.12. Кроме того, X-Server 1.13.3 дополнен патчами, улучшающими систему обнаружения мониторов. Также обновлены видеодрайверы — Intel до версии 2.21.6, NVIDIA 319.17 и NVIDIA304 (для карт < 8xxx) 304.88, AMD Catalyst (fglrx) 13.4. Системные репозитории подключены сразу после установки, благодаря чему пользователь может сразу получать обновления системы.

» Пользователи ROSA Desktop Fresh 2012 получат обновление ROSA Desktop Fresh R1 через штатную систему обновлений.

Новые пользователи могут скачать установочные образы системы с сайта компании «РОСА»: <http://mirror.rosalab.ru/rosa2012.1/iso/ROSA.Fresh.R1/>.

НАЙДИ РАБОТУ ЛЕГКО!

на www.hh.ru



САЙТ РАЗРЕШЁН ДЛЯ ПОСЕЩЕНИЯ БЕЗ ОГРАНИЧЕНИЯ ВОЗРАСТА

hh **ru**
HeadHunter

Выбирай из более чем 200 000 вакансий

НЕМЕЦКИЙ ОПЫТ

Мюнхен «за», а Берлин «пока нет»

Патенты на ПО в Германии могут стать вне закона, а вот продвижение СПО притормаживает.

Сразу две новости пришло из Германии. Первая хорошая, а вторая — плохая. Начнем с плохой.

Парламент Берлина во второй раз отклонил предложение партии Зеленых по переводу государственных учреждений города на открытое ПО. Предложение включало пятилетний план перевода на открытое ПО 25 % компьютеров из 68 000 рабочих станций берлинских учреждений. Предыдущая попытка аналогичного предложения была предпринята в 2007 году и тоже была провалена.

На сей раз козырем авторов инициативы стал успешный опыт перехода на СПО города Мюнхена, но и этого оказалось недостаточно. Представители двух ведущих немецких партий проголосовали против, продвигая собственную инициативу, не предусматривающую кардинальных изменений, но основанную на принципе поддержки открытых стандартов для обеспечения переносимости документов между различными платформами. При этом альтернативное предложение не исключает применение СПО, но вводит единые требования к используемому ПО.

А вот и хорошая новость. Завершив рассмотрение представленной в апреле инициативы против патентов на программное обеспечение, Парламент Германии утвердил предписание правительству предпринять шаги по изменению законо-

ПО — невзирая на патенты, разработчики должны иметь право без ограничений распространять свои программные продукты как свободные. Напомним, что несмотря на то, что патенты на ПО в соответствии с Европейской конвенцией о патентах являются нелегальными, Европейское патентное ведомство продолжает выдавать такие патенты.

А что же в Мюнхене? В конце мая администрация города Мюнхен анонсировала успешное завершение проекта по миграции более 14 000 рабочих станций государственных учреждений на Linux и свободное программное обеспечение. Перевод госучреждений Мюнхена на открытое ПО продолжается с 2003 года и уже позволил городскому бюджету сэкономить более 10 млн евро; при этом экономия средств стала лишь сопутствующим фактором и никогда не рассматривалась в качестве основной задачи. Основной целью миграции было сокращение зависимости от отдельных производителей ПО, обеспечение поддержки открытых стандартов и стимулирование работы местных IT-компаний.

В настоящее время ИТ-департамент оценивает дальнейшую стратегию использования открытого ПО, чтобы убедиться, что город сможет своими силами поддерживать созданную инфраструктуру и обеспечить взаимодействие с граждана-

ми, предприятиями и другими органами государственной власти. Судя по докладу Петера Хофманна [Peter Hofmann], руко-

водителя проекта по миграции города на Linux, стратегия поддержания открытого ПО, подразумевающая, что вместо централизованной ИТ-службы или привлечения сторонних организаций каждый департамент своими силами выполняет сопровождение внедренных решений, будет сохранена, так как проект изначально ориентирован на самодостаточность.

Вместо одного большого глобального преобразования проект миграции на открытое ПО избрал движение маленькими шажками, отдавая предпочтение сохранению качества и самостоятельно созданию требуемых систем (например, проектом разработаны средства для



► Рекомендации по отмене патентов на ПО, принятые в Парламенте Германии, назревали давно.

автоматизации миграции макросов, форм и VB-скриптов в документах, при этом все разработки передавались в сообщество).

На рабочие станции в госучреждении установлен поддерживаемый собственными силами Linux-дистрибутив LiMux, представляющий собой специально адаптированную для использования в муниципальных учреждениях редакцию Ubuntu Linux. Переход на LiMux стал завершающей фазой миграции: на предыдущих этапах был осуществлен переход на *OpenOffice.org* и такие коммуникационные приложения, как *Thunderbird* и *Firefox*. Кроме того, была произведена постепенная замена специфичных приложений на унифицированные web-ориентированные службы, способные работать через браузер, независимо от типа операционной системы. Более сложные приложения были заменены на многоплатформенные аналоги на языке Java.

Сейчас LiMux используется на 14 000 настольных систем. Миграция произведена в 51 учреждении в различных районах города. Не переведенными на LiMux остается около 1000 компьютеров. Таким образом, на Linux перешло 93 % всех рабочих станций, что значительно выше изначально поставленной цели, а именно — перевода на LiMux 80 % всех систем. В качестве следующего шага упоминается переход на более новую редакцию свободного офисного пакета, при этом выбор уже сделан в пользу проекта *LibreOffice*, который заменит собой используемый с 2006 года пакет *OpenOffice.org*. Также запланировано обновление LiMux до нового LTS-выпуска Ubuntu.

«ПО должно подпадать только под действие авторского права.»

дательства с целью ограничения выдачи патентов на компьютерные программы. По мнению Парламента, программное обеспечение должно подпадать только под действие авторского права, и права владельцев имущественных прав на код не должны ущемляться патентами на ПО, принадлежащими третьим лицам. Исключение предусмотрено только для патентов на программное обеспечение, используемое в качестве замены для механических или электромагнитных компонентов.

Кроме того, отдельно оговорено, что связанные с патентами решения правительства не должны отразиться на легальности распространения свободного

softline®



Services

Software

Cloud

ИТ-архитектура вашего бизнеса



ХОД РАЗВИТИЯ

Улучшения и обновления Ubuntu Touch

Разработка Canonical начала обретать свойства полноценной мобильной платформы.

Ubuntu Touch переходит на стадию доработки деталей. Пока неясно, что ждет этот вариант Ubuntu на рынке, но уже невооруженным глазом можно видеть успехи Canonical в процессе разработки. Компания разрабатывает еще около 10 различных проектов (от настольной версии Ubuntu до проприетарного сервиса для сисадминов LandScape).

В свежих ежедневных тестовых сборках Ubuntu 13.10 появилась возможность установки базовых приложений, развиваемых в рамках Ubuntu Touch, мобильной версии дистрибутива для смартфонов и план-

уже работает достаточно стабильно, поддерживает вкладки и отличается перемещением адресной строки и панели инструментов в нижнюю часть экрана. Менеджер фотографий позволяет группировать содержимое фотоархива по событиям, датам, альбомам и отдельным снимкам. Для фотографий могут применяться операции редактирования, такие как кадрирование, вращение и улучшение качества.

Джоно Бэкон [Jonno Bacon], менеджер по взаимодействию с сообществом компании Canonical, опубликовал в своем блоге видеодемонстрацию текущего состояния разработки проекта Ubuntu Phone, при установке на смартфон Galaxy Nexus.

По мнению Бэкона, внесенная в мае

большая порция улучшений уже позволяет применить платформу для повседневного использования энтузиастами, заинтересованными в тестировании Ubuntu Phone. Посмотреть видео можно на сайте <http://www.jonno Bacon.org/2013/06/07/ubuntu-phone-video-demo/>.

Платформа поддерживает прием и отправку звонков и коротких сообщений, позволяет просматривать сайты, общаться в социальных сетях, поддерживает многозадачность.

Почти все ранее используемые заглушки с макетами программ заменены на прототипы рабочих приложений. В программы добавлено формирование интерфейса с учетом ориентации экрана.

Ранее наблюдаемые крахи в процессе работы устранены, система ведет себя достаточно стабильно. Заметно улучшена работа системы управления питанием. Среди проблем отмечаются небольшие задержки в работе, вызванные отсутствием должных оптимизаций, которые будут добавлены позднее. Пока остается неработоспособен GPS.

А Марк Шаттлворт [Mark Shuttleworth] перевел «Ошибка номер 1» в категорию решенных проблем. Напомним, что Ошибка номер 1 в системе трекинга ошибок проекта Ubuntu ссылается на доминирующее положение Windows на рынке пользовательских систем и указывает на то, что данная ситуация требует исправления. 30 мая 2013 года Шаттлворт закрыл ошибку с комментарием, что проблему можно считать исчерпанной и решение найдено — с развитием портативных систем, смартфонов и планшетов, Windows больше не является доминирующей пользовательской ОС: ее вытеснили мобильные платформы и, в частности, открытая Linux-платформа Android.

С 2004 года, когда была заведена запись для Ошибки номер 1, на рынке операционных систем произошел значительный сдвиг в сторону открытости. Даже Microsoft уже не может просто игнорировать движение открытого кода, о чем, например, говорит то, что в облачном сервисе Windows Azure реализована полноценная поддержка запуска гостевых систем на базе Linux.

И при всей неоднозначности отношения сообщества пользователей к инновациям от компании Марка Шаттлворта, нельзя не признать, что Linux сумел шагнуть навстречу широким массам во многом именно благодаря усилиям Canonical.

«В Windows Azure реализована поддержка систем на Linux.»

шетных компьютеров. В настоящее время пользователи обычных десктоп-сборок дистрибутива уже могут протестировать мобильные версии таких приложений, как менеджер фотографий, медиа-плеер, редактор заметок и минималистичный веб-браузер на базе движка WebKit. Браузер



➤ Можно уже сейчас попробовать интерфейс Ubuntu Touch на смартфоне или в эмуляторе для Ubuntu Desktop.

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



Tizen расширяет охват

Усиливается конкуренция на мобильном рынке.

Стремительное развитие мобильных технологий за последние 6–7 лет вынудило патриарха ИТ-индустрии Intel разработать свою мобильную ОС, включившись в гонку.

Плоды трудов инженеров компании уже демонстрируются на выставках и конференциях. На прошедшей в конце мая Tizen Conference 2013 в Сан-Франциско Intel показала прототип ультрабука, поставляемого с платформой Tizen. Пользовательская оболочка Tizen Shell базируется на GNOME Shell. Данный вариант окружения Tizen для ноутбуков войдет в состав будущего выпуска платформы Tizen 3.0, который выйдет за рамки системы для смартфонов. Код наработок Intel для ноутбуков и готовый образ для установки Tizen на выпускаемые ноутбуки планируется вскоре опубликовать.

Оболочка Tizen Shell выполнена как JavaScript-дополнение к GNOME Shell, аналогично реализации классического режима рабочего стола в GNOME 3.8. Свойственные платформе Tizen web-ориентированные компоненты не обошли и версию для ноутбуков. Так, помимо штатных возможностей GNOME 3, интегрирована поддержка виджетов, работающих поверх специальной прослойки для выполнения обособленных web-приложений, написанных с использованием технологий HTML5 и Tizen Web API. Таким образом, приложения, написанные с использованием Tizen Web API, смогут работать во всех вариантах системы Tizen, в том числе на смартфонах и ноутбуках.

Для управления сетью вместо используемого в GNOME конфигуратора *Network Manager* задействован *Connman*, отличающийся низким потреблением системных ресурсов и наличием гибких средств для расширения функциональности через плагины.

Среди приложений в базовой поставке — офисный пакет *LibreOffice*, менеджер фотографий *Shotwell*, медиа-плеер *Rhythmbox*, Linux-клиент для сервиса цифровой дистрибуции игр *Steam*. Для работы с медиа-данными предусмотрены звуковой сервер *PulseAudio* и Bluetooth-стек *BlueZ*.

В пользовательское окружение плотно интегрированы средства поддержки online-служб, включая Gmail/Gtalk, Jabber, IRC, Facebook, Twitter, VoIP/SIP. «Из коробки» доступны средства для создания под-

контрольных облачных хранилищ в стиле Dropbox на базе платформы OwnCloud.

Там же был показан прототип экспериментального пользовательского интерфейса Obsidian, для смартфонов на базе Tizen. Не исключается создание варианта нового интерфейса и для Android. Obsidian будет предложен для использования с эталонной платформой Josephine, ориентированной на создание мобильных устройств на базе процессоров Atom.

Интерфейс оформлен в классическом 2D-стиле с плотной компоновкой элементов. Наиболее заметными отличиями является присутствие трех программных кнопок в нижней части экрана, поддержка режима разблокировки через скользящий жест и визуализация уведомлений в виде вращающейся иконки приложения, для которого есть непрочитанные уведомления.

Samsung также развивает вариант своего фирменного интерфейса TouchWiz для Tizen, но делает ставку на устройства на базе архитектуры ARM. Кроме того, аналитическая компания ABI Research прогнозирует доминирующее положение Tizen в странах Азиатско-Тихоокеанского региона, полагая что Tizen имеет шансы занять до 65 % рынка в данном регионе.

А «лучший друг» Intel, компания AMD, объявила о планах по разработке OEM-решений для планшетов и ноутбуков на базе Android и Chrome OS, в дополнение к уже выпускаемым платформам для Windows 8.

Производители оборудования явно осознают: будущее ИТ-компаний, нацеленных на конечного пользователя, напрямую зависит от мобильных технологий. **LXF**



➤ Интерфейс Obsidian похож на Android, и даже работает с Android.

Новости короткой строкой

➤ Сисадминам: 25–28 июля Softline проводит VIII Всероссийский слет ИТ-специалистов-2013 на знаменитой Поляне рядом с базой отдыха «Лаврова-Песочня» под Калугой. Источник: <http://sletadminov.ru/>

➤ Вышла финальная версия платформы для автоматизации бухгалтерского учета и автоматизации деятельности предприятия «1С:Предприятие» 8.3, в состав которой впервые включены клиентские приложения и инструменты администрирования для Linux. Источник: www.1c.ru

➤ В Эстонии представлена целевая программа по созданию государственных информационных сервисов. Ее особенность — развитие всех ключевых сервисов в виде свободных проектов, доступных в исходных текстах и подразумевающих возможность создания производных продуктов. Источник: joinup.ec.europa.eu

➤ Группа исследователей из Германии обнаружила, что компания Microsoft предположительно занимается анализом содержания мгновенных сообщений, отправляемых через Skype. Источник: www.h-online.com

➤ Компания Фоксхолл и сообщество Mozilla объявили о сотрудничестве по продвижению мобильной платформы Firefox OS. В рамках партнерства будет подготовлена новая серия устройств, предлагаемых OEM-производителям. Источник: blog.mozilla.org

➤ Некоммерческая правозащитная организация Electronic Frontier Foundation (EFF) заявила официальный протест на действия консорциума W3C, начавшего процесс стандартизации средств DRM для HTML5-контента. Источник: www.eff.org

➤ Компания Cisco совместно с итальянским провайдером телефонии Messagenet потребовали от Европейского суда пересмотреть решение по совершенной в 2011 году сделке по покупке корпорацией Microsoft компании Skype. Источник: www.reuters.com

➤ В дополнение к ранее опубликованным исходным текстам варианта ядра Linux, компания Google открыла доступ к образу Android-прошивки для интерактивных электронных очков Google Glass. 331-МБ образ можно установить при помощи Android SDK. Источник: www.opennet.ru

ИТ-награда — ДОСТОЙНЫМ

Завершился V Всероссийский конкурс «Лучшие 10 ИТ-проектов для госсектора 2013».

ИТ-премии ComNews вручаются уже пятый год подряд. Лауреаты получили наградные статуэтки на торжественной церемонии в рамках форума «Умный город будущего».

В жюри конкурса работали: вице-президент ОАО «Ростелеком» Алексей Басов, и.о. вице-президента по инновационному развитию ОАО «Ростелеком» Роман Кравцов, главный редактор ComNews Леонид Коник, руководитель Российского агентства развития информационного общества (РАРИО) Александр Айгистов, координатор конкурса «Лучший свободный проект России» Сергей Голубев, заместитель директора Российской ассоциации электронных коммуникаций (РАЭК) Сергей Гребенников, заместитель генерального директора Фонда содействия развитию МП НТС Павел Гудков, директор ФГУП «ВНИИ ПВТИ» Дмитрий Дунаев, исполнительный директор АРПП «Отечественный софт» Светлана Инфимовская, заместитель председателя правительства Республики Северная Осетия — Алания Леонид Кесельбренер, президент «РУССОФТ» Валентин Макаров, директор департамента информационно-коммуникационных технологий Торгово-промышленной палаты РФ Михаил Максимов, заместитель руководителя ФНС России Андрей Петрушин, руководитель межрегионального информационно-аналитического управления Росздравнадзора Кирилл Поспелов, исполнительный директор РАСПО Юлия Овчинникова, член экспертного совета по Интернету и развитию электронной демократии при Комитете Госдумы по информационной политике, ИТ и связи, коммерческий директор агентства «Деловая Россия» Любовь Орлова, депутат Государственной Думы РФ Роберт Шлегель, директор центра технологий электронного правительства Андрей Чугунов. Основные критерии оценки решений для госсектора, которыми в этом году руководствова-



► Награды для башковитых.

лось жюри, включали инновационность, социальную значимость, устойчивость, масштабируемость и прозрачность, а формат конкурса предполагал награждение как заказчика проекта, так и его исполнителя.

На церемонии награждения в Большом зале Международного мультимедийного пресс-центра «РИА Новости» собралось больше 120 человек, со сцены их поприветствовал директор Департамента развития электронного правительства Минкомсвязи Алексей Козырев. Из его рук и получил награду победитель в первой номинации — «Лучший инновационный проект» — компания «Энвижн Групп». Этот интегратор реализовал проект по созданию интеллектуальных систем безопасности и мультимедийных сервисов для Технопарка города Набережные Челны. «Мы надеемся, что в ближайшее время этот технопарк будет гордостью среди других технопарков России», — отметил Алексей Козырев. «Это честь, потому что проект был очень сложный — и мы закончили его на два квартала раньше, чем предполагалось», — ответил руководитель проекта, директор дивизиона по работе с государственными структурами Александр Мартынов.

В номинации «Лучший межведомственный проект» победителем стала «Фирма АйТи. Информационные технологии» с проектом по развитию и поддержке информационной системы «Мониторинг деятельности контрольно-надзорных органов» для Минэкономразвития. Награду вручил Роман Кравцов («Ростелеком» уже четвертый год подряд поддерживает инициативу проведения конкурса в качестве стратегического партнера).

«Linux Format был информационным спонсором конкурса.»

► Конкурс может похвастаться мощной поддержкой.



Досье ComNews

Конкурс «Лучшие 10 IT-проектов для госсектора» учрежден Министерством связи и массовых коммуникаций и ГК ComNews в 2009 г. и ежегодно чествует заказчиков и исполнителей наиболее значимых и актуальных проектов в области ИТ, обеспечивающих инновации в работе ведомств и государственных структур. За пять лет проведения конкурс стал центром внимания крупнейших федеральных деловых и индустриальных СМИ.

Конкурс привлёк ИТ-специалистов и руководителей из более чем 60 субъектов РФ. В ходе конкурсов были отобраны, исследованы и отмечены десятки решений и проектов в ведущих госучреждениях и ведомствах, например, в МВД, Пенсионном фонде, ФМС, Минюсте, ФНС РФ и др. Некоторые из них портированы в других ведомствах. В ИТ-конкурсе ComNews принимали участие подрядные организации Минфина, Департамента ИТ Москвы, «Ростелекома», Центробанка, «Почты России», «Аэрофлота», многих других госструктур, ведомств и госкомпаний.

В номинации «Лучший проект с использованием СПО» победила компания «ЛАНИТ» с проектом по апробации и доработке программно-технического решения «Информационная система поддержки лицензирования отдельных видов деятельности» и за создание инфраструктуры обеспечения тиражирования решения для Министерства потребительского рынка и услуг Московской области.

Специальным почетным дипломом «За вклад в развитие ИТ в регионе на базе СПО» жюри решило отметить проект группы компаний 2B Group по разработке и внедрению автоматизированной электронной системы «Электронный Барнаул» для городской администрации, который победил в номинации «Лучший проект «Умный город»».

В номинации «Лучший проект в образовании или здравоохранении» победила компания Helios Net (ЗАО «Интерактивные системы»). Компания реализовала комплекс работ по организации доступа в Интернет детям с ограниченными возможностями и педагогическим работникам для Центра психолого-медико-социального сопровождения «Ресурс» Свердловской области.

Оргкомитет конкурса не обошел вниманием проект компании «ЛАНИТ» и Департамента ИТ г. Москвы по разработке Единой медицинской информационно-аналитической системы (ЕМИАС) города Москвы для Департамента здравоохранения столицы. «Команда из 150 разработчиков поучаствовала в сохранении здоровья четырех миллионов москвичей», — заявил собравшимся замруководителя ИТ-департамента Владимир Макаров.

В номинации «Лучший стартап на службе государства» победителем стала группа компаний «Техносерв» с проектом по созданию автоматизированной информационной системы «Государственный лесной реестр» для Федерального агентства лесного хозяйства.

Победитель в номинации «Лучшее платежное решение» — «Фирма АйТи. Информационные технологии» с проектом по оказанию услуг на поставку, монтаж и пуско-наладочные работы автоматов продажи билетов на одну и две поездки на картах «Ультралайт» для ГУП «Московский метрополитен».

В номинации «Лучшее мобильное решение» победило ФГУП «Главный научно-исследовательский вычислительный центр Федеральной налоговой службы» с проектом по разработке программного обеспечения, позволяющего формировать заявление на регистрацию физического лица в качестве индивидуального



➤ Алексей Козырев приветствует победителей.



➤ Награждение Федерального агентства лесного хозяйства (справа) и ГК «Техносерв» (в центре).

предпринимателя в интерактивном режиме через Интернет, в том числе с мобильных устройств.

«Лучшее решение для колл-центра», по мнению жюри, внедрила компания «Бюджетные и финансовые технологии» в рамках проекта «Внедрение автоматизированной системы «Капелла» в Региональном центре телефонного обслуживания Республики Саха (Якутия)».

В номинации «Лучшее облачное решение» победила компания «Адекват», поставившая систему электронного документооборота учреждений здравоохранения Томской области.

Специальной наградой в номинации «Самое эффективное решение для контакт-центра» жюри отметило проект компании «Голси» по разработке и внедрению программно-аппаратного комплекса для автоматической обработки вызовов «Аэрофлот-Инфоцентр».

Кроме того, специальной наградой жюри конкурса отметило проект «Российская общественная инициатива» как лучший онлайн-проект в сфере развития гражданского общества.

Наконец, заслуженно отметил достижения интеграторов и госструктур в сфере информатизации, жюри наградило почетным дипломом лидера телекоммуникационной отрасли — компанию «Ростелеком» за вклад в информатизацию государственных органов РФ. **LXF**

«Критерии оценки включали социальную значимость.»



➤ Компания ГОЛСИ разработала самое эффективное решение для колл-центров.

Новинки программного и аппаратного обеспечения в описании наших экспертов



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

Песнь о потомках Sid'a

Жил-был мальчик Sid, который все портил. Да так умело, что его именем назвали unstable-ветку дистрибутива Debian. Которая, как знают его поклонники, может дать фору иным стабильным. А потому издревле многие хотели применить ее в мирных, а не только в тестировочных целях. Что и было реализовано в виде дистрибутива sidux, который по копирастическим причинам сменил имя на aptosid, а затем ответил от себя дистрибутив siduction.

При внешнем сходстве, между этими дистрибутивами есть и существенные различия. aptosid — это действительно Sid, адаптированный для практического десктопного применения, по умолчанию — со средой KDE. Модель его разработки — полный rolling, то есть штатно предлагает не апдейт с релиза на релиз, а перманентное обновление системы вслед за таковым исходного Sid'a.

Сборки siduction охватывают почти все рабочие среды — от тяжелых (KDE, Gnome 3) через «средние» (Xfce) до легких (LXDE, Razor-qt) и noX («голая» консоль). И все они регулярно обновляются до текущих версий соответствующего десктопа (как, кстати, и ядро), иногда даже опережая прародительский Sid. Модель разработки siduction ближе к регулярным релизам, нежели к rolling'у.

Это я к тому, что, несмотря на все поползновения разделить мир Linux на два антагонистических класса — systemd'овцев и ubuntu'йцев, он пока не скудеет разнообразием: не далее как в мае вышли новые релизы и aptosid, и siduction. Что и радует.

alv@posix.ru

Сегодня мы рассматриваем:

- Ubuntu 13.04** 16
Ubuntu почти что пинком вышибли из его гнездышка на вершине рейтингов, но каждая его новая версия вызывает больше волнений, чем выходящие вслед за ней релизы нового фаворита — Mint. Может, у них и не всегда все правильно, но зато всегда интересно.
- Ciseco XinoRF** 18
Удаленный контроль — это грандиозно. Легко вообразить, как ликовал наш Бен, заполучив в свои руки плату совместимости с Arduino, снабженную радиосвязью. Ко всему хорошему, она запросто осваивается, в ее названии красуется буква X — для баллов умника, а дальность ее действия (заявленная) — 300 м!
- Manjaro Linux 0.8.5** 19
Что такое Skippy? Возобновляемый, остросовременный релиз дистрибутива Linux, с несложной методикой установки и дружелюбным интерфейсом пользователя? И где ж он раньше-то отсиживался?
- Kerbal Space Program** 20
О том, что физика может быть удовольствием, похоже, знает лишь узкий круг элиты. Но этот симулятор освоения космоса должен принести широким массам благую весть о гравитации, ракетах и маленьких зеленых человечках.
- OPPO Find 5** 21
Этот китайский смартфон-новичок покусается вдрызг расколотить черные тыквы — Apple и Samsung. Он представляет весь джентльменский набор современного коммуникатора, с качеством не хуже, чем у «больших», и притом за две трети их цены.



› Зря Ubuntu назвал эту версию Педантичный Панголин, а не Розовый Пони.



› Прежде чем слать в космос кербалов, поищем признаки разумной жизни на Земле.

Сравнение: Дистрибутивы для планшетников с. 24

Android x86



Fedora



Kubuntu Active



OpenSUSE



Ubuntu



Поглядим, как любимый Linux совершил кульбит с клавиатуры на сенсорный экран, и разберемся, стоит ли эта ваша новая парадигма рабочего стола поднятой вокруг нее шумихи.



Ubuntu 13.04

Вновь открыв для себя дистрибутив, знакомый ему много лет, Грэм Моррисон обнаруживает немало перемен.

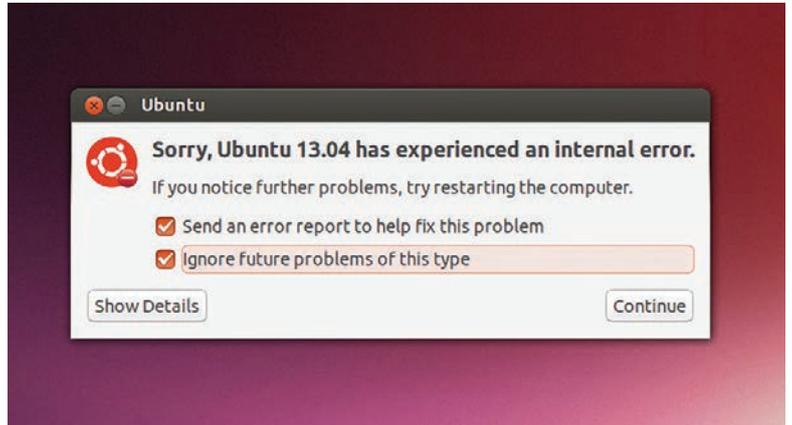
Вкратце

» Дистрибутив, официально известный как Самый Популярный в Мире. См. также: Mint, Mageia, Fedora.

Ubuntu есть за что полюбить. Для начала, это дистрибутив с самым простым и красивым процессом установки, где обновления скачиваются по умолчанию, а сторонние пакеты можно установить, ответив всего на пару вопросов. На домашнем компьютере, ноутбуке, ультрабуке или на сервере — ваш образ действий одинаков: те же файлы настройки, ни с чем не сравнимое богатство пакетов, традиция качества и стабильности. И все задаром. А еще, этот дистрибутив, одним из немногих, отваживается ломать стереотипы и идти новым путем.

В итоге Ubuntu никак нельзя обвинить в том, что он топчется на месте. И это хорошо: для ожидаемых обновлений есть openSUSE и Fedora. Но — куда же без «но» — и за качество, и за прорыв нужно платить. О цене и сопряженных конфликтах подробно говорится в нашей главной статье. Но для пользователей все в основном сводится к тому, что Ubuntu все дальше уходит от идеального образа Linux-дистрибутива и того, чем сам Ubuntu был раньше: дистрибутивом на базе Debian, ориентированным на сообщество. Теперь же его разработчики больше экспериментируют в свое удовольствие, будь то создание рабочего стола Unity или предпочтение растущему графическому серверу Wayland собственного Mir.

На первый взгляд создается впечатление, что Ubuntu стремится сделать свой дистрибутив самым простым, интуитивным



» Даже если приложения падают, Ubuntu разумно спрашивает, отображать ли сообщения об ошибках впредь.

и «не-линуковским», чем и объясняется его спорность. Но пример Android подтверждает, что делать «не как в Linux» — это лучший путь к успеху.

Легенда о фениксе

Неплохо, однако, что процесс установки в 13.04 остался прежним. Сохранить разбивку в 'btrfs' из Ubuntu 12.10 не удалось, но в плане установки Ubuntu по-прежнему нет равных. Нам понравилось, что сохранена опция разметить диск вручную, самому определить, где жить загрузчику. В менее дружелюбных к новичкам дистрибутивах эта опция нередко отсутствует, а ведь она незаменима в мультизагрузочных ПК с несколькими ОС.

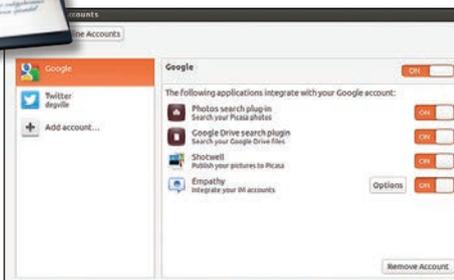
Не прошло и 20 минут, как перед нами во всем блеске предстал новый рабочий стол 13.04. Unity, нынешнее рабочее окружение Ubuntu, улучшается с каждым релизом и составляет костяк новой экосистемы Ubuntu, которая обещает привести к единообразию интерфейс на телефонах, планшетах и телевизорах Ubuntu (если таковые появятся).

Не впадая в полемику о Gnome 3 против Unity, скажем, что Unity все же стал полезным и оригинальным рабочим столом, а 7-я версия ощущается быстрее и стабильнее прошлых релизов. Заметных изменений немного, но они важны. Видеоизменились несколько иконок, добавилась функциональность. Для перебора окон теперь можно использовать прокрутку мышью, а у некоторых приложений обогатились контекстные меню. На рабочем столе теперь в призрачной форме указывается, где окажется окно после перемещения; есть и другие визуальные эффекты.

Покритиковать можно лишь то, что иконки и фон придают Ubuntu несколько устаревший вид — уж очень долго не может арт-группа завершить оформление. Как и со многим другим в Ubuntu, технологическая часть великолепна, а вот пользовательская выглядит недоработанной.

Прекрасный тому пример — портал Dash. Он становится невероятно прочным мостом между локальными файлами и интернет-контентом — быстр и прост в использовании, а разнообразные линзы, помогающие поиску, быстро совершенствуются. Но поле поиска должно бы

Свойства навскидку



Улучшен контроль учетных записей

При управлении учетными записями и их интеграции с рабочим столом теперь можно отключать и подключать только нужные; это облегчает работу пользователей.



Новая линза Социальных сетей

Предоставляет быстрый доступ к записям и обновлениям социальных сетей вроде Twitter и Facebook, экономя ваши усилия на проверку их вручную.

переноситься с одной линзы на другую. Например: откройте Dash с помощью клавиши Windows и наберите 'Steam'. Если клиент Steam установлен, вы сможете запустить его прямо из результатов поиска. Если нет, нужно перейти в линзу приложений (Ctrl+Tab). Но стоит вам это сделать, как текстовое поле очищается и заменяется тем, что вы в последний раз искали в данной линзе. Не очень-то интуитивно. И было бы хорошо, если бы Dash всегда возвращался к последней использованной линзе. По нашему мнению, такие мелочи бросаются в глаза, когда речь заходит о достижении подлинно профессионального качества использования, и мы считаем, что Ubuntu пошло бы на пользу объективное пользовательское тестирование и обратная связь.

Сама по себе линза приложений великолепна: привыкнув искать программы таким способом, вернуться к привычным пакетным менеджерам очень трудно. Стоит начать вводить название, и внизу тут же появляется иконка. Наведите курсор, чтобы выбрать пакет, и получите краткое описание и экранный снимок. Снова нажимаете Enter, и вот все установлено и добавлено на панель — идеально! Чудесный способ окупиться в свободное ПО и откровение для мигрантов с Mac или Windows, привыкших к урезанным условно-бесплатным или запретительно дорогим приложениям.

Любимый рабочий стол

Рабочий стол, по ощущениям, тоже фантастика. В отличие от Gnome, приложения открываются не на весь экран. Это делает Ubuntu гораздо привычнее. Перенесение меню на верхнюю панель тоже становится более осмысленным. В предыдущих

Дебри Amazon'a

В линзе приложений проблем с рекламной информацией не возникало, чего не скажешь об отображении результатов поиска на Amazon через линзу основного поиска. Проблема в том, что она неуместна, отвлекает и нарушает приватность, хотя, как выяснилось, данные поиска в Ubuntu теперь отправляются в зашифрованном виде сначала на собственные серверы, а уже потом на Amazon.

Неясно, почему бы было не создать для Amazon свою линзу и пользоваться ею отдельно, но самое вопиющее — что нет опции для отключения результатов с Amazon, кроме ручного удаления пакета. Лучшее, что Ubuntu может предложить, это «Отображать результаты поиска в Сети» в панели настроек Приватности. Но отключив ее, вы также лишаетесь и таких потрясающих возможностей, как поиск по Google Docs. Технические препятствия для добавления этой функции нет — в ранних макетах панели Приватности была даже опция «Отображать коммерческие предложения от партнеров Ubuntu».



В отличие от 12.10, нам не удалось отсеять некорректные результаты поиска в Amazon.

Нечто подобное требуется и для тонкой настройки интеграции Google в панели Online Accounts, вместо того, чтобы отдавать пользовательский поиск на откуп Amazon. Это главная проблема Ubuntu 13.04.

релизах картину портило то, что часть приложений сохранили старое меню, а часть была Ubuntu'изирована. Теперь же можно быть практически уверенным, что после установки приложений на базе GTK, вроде GIMP, Inkscape, Scribus, Firefox, LibreOffice и Tomahawk, проблем не возникнет. Блестящие результаты достигнуты с внутренней интеграцией. С добавлением Google в менеджер Online Account клиент сообщений Empathy стал автоматически настраивать Google Chat, и появился более тонкий контроль над отдельными сервисами Google.

Не обошлось и без сбоев. Постоянно падала панель Online Accounts, и изредка вылетал Firefox. Однако после повторного падения детектор сбоев предложил вариант

игнорировать подобную ошибку в дальнейшем, и это полезная функция.

Но вернемся к плюсам. Ubuntu движется к успеху. С точки зрения новичка, в ее интерфейсе нет ничего «специфически Linux'ового». Видеть панель запуска слева привычно как пользователям Windows, так и OS X — особенно тем, кто не хочет, чтобы на них обрушился Windows 8. Диапазон предлагаемых приложений просто невиданный, аппаратная поддержка великолепна, активное сообщество готово помочь. Steam является официальным партнером Ubuntu, можно купить машину с предустановленной системой от Dell, а вскоре Unity будет доступен на телефонах и планшетах. Ubuntu нужно оценивать по конечному результату, а не политике, и мы считаем, что он побеждает. Что же утратил Ubuntu? А ничего. Но это уже не тот Ubuntu, что пленил нас в 2004. И, с учетом почти десяти истекших лет, оно и к лучшему. **LXF**

Ubuntu и смотрит-ся отлично, и предлагает уникальные функции.



LINUX FORMAT Вердикт

Ubuntu 13.04

Разработчик: Canonical Ltd.
Сайт: www.ubuntu.com
Лицензия: Разные свободные лицензии

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

Ubuntu предлагает невероятное удобство по отличной цене — безвозмездно, то есть даром.

Рейтинг 9/10

Ciseco XinoRF

Бен Эверард дорвался до нового дистанционно управляемого Arduino.

Вкратце

» Плата со встроенной радиосвязью, совместимая с Arduino Uno. См. также: Arduino с XBee или Wi-Fi-шилдом.

Arduino — это программируемая плата микроконтроллера, которую можно использовать для создания аппаратных проектов, и одно из самых успешных открытых устройств. Помимо собственной популярности, он также стал основой для множества других подобных плат, создатели которых подработали исходный дизайн под собственные нужды.

XinoRF — один из таких. Он совместим по выводам с Arduino Uno R3, так что любые расширения (известные как шилды [shields]) должны работать из коробки, как и любой код для родительского устройства. Единственное изменение в конструкции — кнопка перезагрузки, повернутая на 90 градусов, чтобы до нее было проще добраться при наличии шилдов, а также мини-USB, вместо стандартного разъема USB типа B. Ничто из этого не должно повлиять на ваши проектные решения, если только вы не используете плату каким-нибудь уж очень особым образом.

Главной особенностью XinoRF является SRF-радио. Это крошечное устройство смонтировано непосредственно на плате, но не как шилд, и потому не составляет проблем при наличии ограничений в габаритах или весе. Никакой пайки не требуется. Антенна встроена прямо в схему, без всяких лишних болтающихся проводов.

«Простейший способ получить беспроводную связь с Arduino.»

роительство смонтировано непосредственно на плате, но не как шилд, и потому не составляет проблем при наличии ограничений в габаритах или весе. Никакой пайки не требуется. Антенна встроена прямо в схему, без всяких лишних болтающихся проводов.



» SRF-приемник в XinoRF настолько мал, что его и не заметишь. Расположен он слева, между контроллером и контактами.

На нашей плате сразу было установлено все для беспроводного программирования, так что мы вставили SRF-карту (она продается отдельно) в USB-порт, включили XinoRF и могли загружать наши эскизы в рабочем порядке, как в обычный Arduino. Радио ведет себя так же, как и проводное соединение, и можно передавать и получать данные, используя

```
Serial.println("message");
Serial.read();
```

Естественно, можно подключать более двух устройств: и несколько Xino, и других, использующих совместимые приемники. На сайте Ciseco представлено множество радиомодулей. Помимо тех, что мы использовали, в формате USB и на базе Arduino, вы найдете продукты для Raspberry Pi, платы под XBee и прочее. Взаимодействовать друг с другом одновременно они могут через Local Lightweight Automation Protocol (LLAP), обеспечивающий простую систему адресации.

Там, вдали

Для беспроводных устройств главное — диапазон. Ciseco предлагает несколько совместимых друг с другом радиомодулей, причем SRF — наименее мощный. Мы протестировали соединение между двумя Xino в офисе со множеством разграждений, плюс компьютер на уровне пола. Последовательное соединение начало хромать примерно с 10 метров. Согласно производителю, в идеальных условиях диапазон

составляет 300 м, и хотя в это верится, мы не стали бы поручать ему ответственные соединения. Для больших расстояний Ciseco предлагает модули ARF, которые, как заявлено, охватывают более 1000 м.

XinoRF — простейший и самый выгодный способ получить беспроводные возможности в среде Arduino, и мы надеемся внедрить его в наших проекты. Гораздо больше документации и кода имеется для ZigBee/XBee, и многим новичкам они сперва покажутся привлекательнее, особенно таким, кто ориентируется скорее на работу по инструкции, чем на создание собственных проектов с нуля. Однако для простых коммуникаций XinoRF подойдет даже тем, кто не очень ладит с техникой. LXF

Свойства навскидку



SRF-радио

Прекрасная связь с другими устройствами в коротковолновом и среднем диапазоне.



Совместимость с Arduino

Контроллер ATMEGA238 гарантирует полную совместимость с Arduino Uno.

LINUX FORMAT Вердикт

XinoRF

Разработчик: Ciseco
Сайт: <http://shop.ciseco.co.uk>
Цена: £ 30

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

» Простейший способ добавить к проекту Arduino беспроводное радио.

Рейтинг **9/10**

Manjaro Linux 0.8.5

Дистрибутив, обещающий и новейшее ПО, и дружелюбный GUI? Так ли это, выясняет **Майк Сондерс**.

Вкратце

» Дистрибутив на базе Arch; ориентирован на удобство и доступность для пользователей. См. также Chakra, ArchBang.

Простота, надежность, современность — сочетание этих трех качеств есть Священный Грааль дистрибутивов Linux. Надежных и мощных дистрибутивов-конкурентов масса, но для достижения должной устойчивости они избегают самых свежих версий ПО. С другой стороны, дистрибутивы, специализирующиеся конкретно на последних релизах, сложнее в использовании и иногда падают. Скажем, Arch Linux — прекрасный дистрибутив для опытных пользователей, обеспечивающий быстрый доступ к новому ПО, но это не означает его благосклонность к новичкам, а порой кое-что и ломается.

Высокая цель Manjaro — предоставить все лучшее: плавающий релиз дистрибутива, основанного на Arch, но со всем дружелюбием GUI — то, чего ждет не совсем продвинутый пользователь Linux. Подобные попытки были и раньше, но Manjaro впечатляет особенно: он рвется к лидерству, у разработчиков есть четкое видение проекта, а их презентации (как в дистрибутиве, так и на сайте) тщательно выстроены и профессиональны. Как ни крути, первое впечатление важно.

Доступны три версии: загрузка 1,2-ГБ ISO дает рабочий стол Xfce с полезным набором ПО; более легкий, на базе Openbox, и оснащенный меньшим количеством пакетов, весит порядка 700 МБ; а мини-вариант с весом примерно 500 МБ — чисто базовая система для настройки пользователем. Мы тестировали версию Xfce, оснащенную Firefox, Thunderbird, VLC, GIMP и LibreOffice (компоненты Writer и Calc).



» Manjaro выглядит мило, однако величина кнопки уменьшения окон немногим больше пикселя, так что подцепить ее почти невозможно. Это нас озлило.

Графический установщик Manjaro основан на Linux Mint, с которым обычно очень легко работать. Тем не менее, была одна трудность во время разбиения: устанавливая на машину, где уже был дистрибутив, мы надеялись на опцию быстрых «очистки диска и авто-распределения разбиения», но вместо этого нам пришлось выполнить ручную разбиение, присвоение точек подключения и т.д. Нам это было не сложно, но далекий от техники пользователь мог бы тут и застрять.

Конкурент Arch

После инсталляции произойдет быстрая загрузка в общем яркого и стильного дистрибутива. Хоть он и основан на Arch, но репозитории ПО у него собственные, современные и с хорошим наполнением: там и непонятные игры, и приложения, которые мы обычно добавляем после инсталляции. Также можно брать ПО из Arch Users Repository. Manjaro хвастает собственным менеджером пакетов Pacas, работающим вполне достойно, но особо здесь кичиться нечем. То же касается конфигуратора Manjaro Settings Manager.

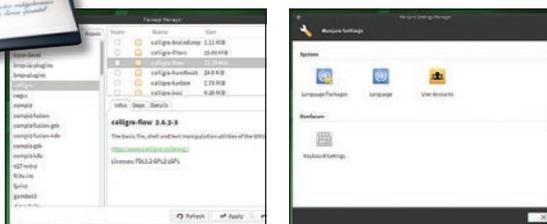
Как во всех случаях, когда дистрибутив носит имя уже имеющегося, документационная ситуация Manjaro запутана. Здесь применимы большинство руководств и рекомендаций, написанных для Arch Linux, но дистрибутивы довольно сильно различаются, и это может вызвать путаницу. Зато команда Manjaro сделала блестящее,

красочное и приветливое «Руководство начинающего пользователя» в формате PDF. Разработчики дистрибутивов часто оставляют такое на потом, но парни из Manjaro действительно думают наперед.

В конечном итоге, Manjaro — впечатляющий и многообещающий дистрибутив, где есть все, чтобы суметь выбиться в «важные птицы». Его домашние инструменты нуждаются в некоторой доработке, но главнейший вопрос будет — насколько хорошо здесь реализован подход плавающего релиза, предоставление пользователям регулярных обновлений без регулярных обломов. Время покажет; а сейчас это привлекательный и доступный вход в мир Arch Linux. **LXF**



Свойства навскидку



Pacas

Менеджер пакетов Manjaro оформлен скупко, а должен бы кричать о доступном через него отличном ПО.

Settings Manager

Settings Manager тоже выглядит на этом этапе довольно скучно, изменить можно всего несколько опций.

LINUX FORMAT Вердикт

Manjaro Linux 0.8.5

Разработчик: Команда Manjaro
Сайт: www.manjaro.org
Цена: Бесплатно под GPL

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

» Пока это не долгожданный дистрибутив мечты, но станет им, если будущие обновления пойдут гладко. Стоит проследить.

Рейтинг 8/10

Kerbal Space Program

Алекс Кокс глотает стероиды, надевает гермошлем и забрасывает жестянку с зелеными человечками на орбиту. И все за один рабочий день.

Вкратце

» Умный и уникальный тренажер освоения космоса типа DIY [Do It Yourself — сделай сам]. Также стоит рассмотреть, но лишь смутно: *Oolite*, *Vega Strike*.

Легко заподозрить, что игра про космические приключения с участием прозорливых, скоропортящихся людей не будет столь привлекательна, как игра про славных зеленых кербалов. Анимированные лица храброй команды — элемент шарма *Kerbal Space Program*, и переправлять одного такого с родной, подозрительно похожей на Землю планеты под названием Kerbin, на поверхность другого объекта — с ближайшего спутника Mup на самую дальнюю, плутоноподобную планету-карлик Eeloo — чрезвычайно увлекательно. Особенно если вы не намерены вернуть своих злополучных кербалов домой. Личная привязанность обусловит новый полет вслед за первым, поскольку придется пересесть на корабль побольше и отправиться спасать команду из отдаленного уголка Солнечной системы.

За слегка комедийным покрытием стоит довольно серьезное моделирование. Чтобы удалось куда-то добраться, не обойтись без серьезных астрономических знаний и математических талантов. Ваша первая вылазка в космос пройдет в изучении периапсиды и апоапсиды, прямого и обратного движения планет, загадочной и очень важной Δv [приращение характеристической скорости, — прим. пер.], хитростей не только гравитации и атмосферы, но и довольно сложных навигационных систем *KSP*. Даже если установлены удобные модули, как например, сильно рекомендуемый модуль автопилота MechJeb, этап первоначального обучения довольно непросто.



» Есть в этом что-то действительно греющее душу, когда врубаешь двигатели на полную и даешь залп.

Количество запчастей в собранном агрегате *KSP* дает огромный потенциал для всех видов летательных аппаратов. Сперва это будет ракета несложной конструкции, с запасом топлива и камерой сгорания, несколькими стабилизаторами и острой носовой частью. Чтобы выйти на орбиту, вам понадобится большой, многоступенчатый летательный аппарат для полетов при различном атмосферном давлении, с увеличенным весом. Однажды зайдя так далеко, все обломки можно оставить на орбите Kerbin и начать путешествие к планетам и спутникам ее системы.

Но это лишь вершина айсберга. Возможно, придется собрать космолет для возврата домой или части космической станции — стыковать на орбите. Ограничения, диктуемые аэродинамикой и гравитацией, означают, что реалистично забросить эти вещи в космос за один присест не выйдет. В более дальних путешествиях уже нужно больше внешнего оборудования, топливо для успешного старта и даже корабль, способный собирать внеземные ресурсы. И кстати, почему бы не отправить на Mup вездеход, позволив кербалам немного побездельничать. *KSP* — это игровая площадка, ограниченная только вашим воображением и инженерным мастерством преданной команды кербала.

Но — будьте готовы к переменам. *KSP* сейчас находится в альфа-стадии разработки: это часть превосходной программы Early Access от компании-разработчика

Valve, значит, можно тут же купить игру, получать все обновления (исключая DLC) и развиваться самому по мере их выхода. Ряд небесных тел на орбите добавился недавно, намекая на план выхода за пределы игровой площадки одной звездной системы, что обычно всем очень нравится.

Игра не выглядит изобилующей ошибками, но ваш опыт может отличаться. Мы сделали ее обзор еще до начала основной кампании — растущее сообщество *KSP* выделило игру из толпы. Она из серии тех, с которыми возятся и пишут о них романы, что люди часто и делают. Наличие шансов самому построить достойный корабль говорит о том, что истинный потенциал игры еще не раскрыт. **LXF**

Свойства новскидку



Построй

Можно легко построить ракету, соединяя друг с другом сотни предлагаемых деталей.



Разрушь

Управлять ракетой непросто, особенно если при конструировании игнорировали аэродинамику и баланс.

LINUX FORMAT Вердикт

Kerbal Space Program 0.19

Разработчик: Squad
Сайт: <http://kerbalspaceprogram.com>
Цена: £17,99 через Steam

Игровой процесс	10/10
Графика	8/10
Продолжительность	10/10
Оправданность цены	7/10

» По прошествии этапа обучения *KSP* становится коробкой с веселыми путешествующими в космосе игрушками. Вам понравится!

Рейтинг **9/10**

OPPO Find 5

Артем Зорин побывал на презентации нового «убийцы флагманов» — и вернулся довольный.

Вкратце

» Смартфон с Full HD-экраном и четырехъядерным процессором по цене гораздо ниже ожидаемой.

Индивидуальность смартфона — свойство ценное: не все потребители желают сливаться с массами владельцев «единообразных» аппаратов Apple, Samsung и других именитых компаний. Поэтому радуется, что к нам стали заглядывать «свежие» производители. Среди них — китайский бренд OPPO.

По своим характеристикам аппарат OPPO Find 5 готов на равных поспорить за наши кошельки и с Samsung Galaxy S IV, и с Sony Xperia Z, и с любым «тяжеловесом» первой половины 2013 года. OPPO отложила в сторону эксперименты с дизайном, сделав выбор в пользу «универсальной» строгой внешности — прямые линии, практически без «игривых» скруглений; смотрится благородно, «на все деньги». Первым делом обращает на себя внимание фронтальная металлическая пластина (нержавеющая сталь), заходящая на верхнюю и боковые стороны. Тыл оставлен под защиту матового пластика, а «лицо» смартфона укрывает слой стекла Gorilla Glass 2 — поцарапать экран вряд ли получится даже умышленно.

Несмотря на внушительный 5-дюймовый экран, смартфон не вызывает ассоциаций с «лопатой». При габаритах 142×69×8,9 мм он хорошо лежит в ладони, да и вес в 165 г рук не оттянет.

Модная величина диагонали экрана топковых смартфонов плавают приблизительно в районе 4,7 дюйма, но с разрешением более чем строго: ни шагу назад, только Full HD (1920×1080 точек). 5-дюймовый IPS-дисплей OPPO Find 5 не имеет воздушной прослойки над сенсорным слоем. При просмотре HD-видео изображение выглядит так живо, что чуть ли не выпрыгивает из экрана. В стане элитных камер обязательно наличие «глазка» на 13 Мп. Оптика OPPO Find 5 позволяет делать фото на скорости 20 снимков в секунду, а видео — 120 кадров в секунду. Аппаратная поддержка HDR-съемки обеспечивает приличное качество даже при дефиците освещения.

40 % из 1400 инженеров завода OPPO отвечают за программную сферу, поэтому Android 4.1 в OPPO Find 5 закономерно подвергся индивидуальной настройке, в основном по визуальной части. Переделана «оболочка» практически всех стандартных приложений: это знакомый Android, но без привнесшегося дизайна



» Очень привлекательный дизайн и цена, которая буквально ошеломляет.

«защитных» программ. Кстати, не за горами обновление до версии 4.2.

«Железки» OPPO Find 5 — точная копия таковых в Sony Xperia Z, и это скорее комплимент. В арсенале смартфона 4-ядерный процессор Qualcomm Snapdragon S4 Pro APQ8064 с частотой 1,5 ГГц, «графика» Adreno 320 и 2 ГБ ОЗУ. «Бумажные» характеристики OPPO Find 5 оправдали сполна, претензий к производительности или плавности работы не возникает. За аппаратное раздолье «отдувается» аккумулятор на 2500 мАч, в среднем на сутки работы. Для хранения информации предлагается 16 или 32 ГБ — в зависимости от версии аппарата. Слот для карт памяти отсутствует — но тому есть убедительное объяснение: опрос, проведенный компанией в Китае, Таиланде и США, показал, что большинству хватает встроенной памяти. Оригинально выполнен слот под SIM-карту. Чтобы извлечь крепление с карточкой, в близлежащее отверстие на корпусе требуется вставить тонкий штырек наподобие разогнутой скрепки.

Музыке на OPPO Find 5 позавидует и дорогой смартфон. Встроенная технология Dirac HD пришла сюда из мира аудиосистем класса Hi-End, и команда Dirac лично поучаствовала в разработке проводной гарнитуры из комплекта смартфона. Итог: эпитеты, касающиеся впечатлений от прослушивания на OPPO Find 5 композиций с высоким битрейтом, только похвальные.

«Беспроводная подборка» у OPPO Find 5 выдалась под стать флагманскому положению: привычные Wi-Fi и Bluetooth (4.0) дополняют NFC и Wi-Fi Display/

DLNA. Последнее означает, что изображение с экрана модели может передаваться на телевизор или другой смартфон либо планшет. Для навигации используется гибридный приемник ГЛОНАСС/GPS, что должно минимизировать вероятность потери связи со спутниками.

OPPO Find 5 интригует прочным корпусом, отличным экраном (кстати, производства японской Sharp), впечатляющей аудиосистемой и многими другими особенностями. Мы нередко видели интересные новинки, чья привлекательность терялась за слишком высокой ценой. Так вот, смартфон OPPO обойдется в 19 990 рублей за версию с 16 ГБ — почти на 10 тысяч дешевле всех основных конкурентов. Желающим заполучить 32 ГБ памяти придется заплатить 22 990 рублей. Весьма заманчиво, не так ли? **LXF**

LINUX FORMAT Вердикт

OPPO Find 5

Разработчик: Компания Орро (Китай)
Сайт: www.oppo.ru
Цена: 19990 или 22990 руб., в зависимости от объема памяти

Функциональность	9/10
Производительность	9/10
Удобство использования	9/10
Оправданность цены	10/10

» Редкое сочетание первоклассного качества и неожиданно невысокой цены.

Рейтинг 9/10



ПЕРВАЯ В РОССИИ

Наш новый MATE

В Москве прошла первая выставка-конгресс MATE-2013.

30 и 31 мая в московском Центре дизайна ARTPLAY прошла первая в России конгресс-выставка Mobile Application & Technology Expo (MATE) по бизнесу в мобильных сетях. Более 800 специалистов констатировали: рынок мобильных приложений в России ожидает бурный рост.

Компания Смайл-Экспо организовала четыре эффективных формата общения: выставку, конференцию, площадку Speed Dealing, павильон карьеры "MATE CAREER". Состоялось более 50 докладов по различным аспектам: от тенденций рынка до монетизации и продвижения.

За 2011 год продано 1,8 млрд мобильных устройств, на сумму свыше \$1 трлн. К 2015 году рынок устройств, осуществляющих связь с Интернетом, составит почти 4 млрд штук. Смартфоны далеко опередили компьютеры по продажам. Традиционные социальные сети уступают место мобильным приложениям, потому что:

- » Интернет на них дешевле стационарного.
- » Сами устройства доступнее.
- » Все больше людей воспринимают свой телефон как часть себя.
- » Пользователи предпочитают делиться моментами жизни прямо здесь и сейчас.
- » Налицо взрыв сверхновой мобильной рекламы.

Спикерами выступили А. Носов (Avia-sales.ru), М. Малый (Mail.ru), П. Плотников и В. Медведев (Dr. Web), Д. Маслеников (Лаборатория Касперского), М. Гакер (QIWI), Р. Алиев (РБК Money), Ю. Долженко (Google Inc.), И. Костров (MobiAds), А. Сидорина (Dunatic Pixels) и многие другие.

Докладчики осветили общие тенденции на рынке мобильных устройств и технологий, в т. ч. мобильную рекламу; вирусное продвижение игр; инструменты мобилизации электронной коммерции; мобильные сервисы оффлайн-магазинов; мобильные приложения в корпоративном и банковском секторе, новости Android. Посетители имели возможность пообщаться со спикерами и задать им вопросы.

На выставке наибольшей популярностью пользовались стенды российских компаний PlayDisplay, HelpLine, Polonium-

» Несмотря на летнее затишье на рынке мобильных технологий во всем мире, дела на MATE кипели.



arts, 2Gis, Unicume, Eski.mobi, Mobiads, Essential Commerce, My-apps, Get Taxi. Участники выставки представили широкий спектр мобильных новинок:

- » решения для интернет-коммерции: Cardomat, MySiteCalls.me, iSpeakVideo, MarketBook (Essential Commerce);
- » проектирование, разработка и тестирование мобильных устройств (Poloniumarts);

«MATE оказался хорошей платформой для демонстрации идей.»

- » ПО, медиа-презентации и digital-журналы для смартфонов и планшетов (Tamzap);
- » крупнейшая сеть рекламы (Mobiads);
- » онлайн-конструктор My-apps, позволяющий без особых навыков, буквально за пару щелчков, создать свое собственное мобильное приложение;
- » инновационный сервис заказа такси для смартфонов Get Taxi;
- » разработка картографического ПО Navitel, а также сервисов автомобильной навигации для России/СНГ, Европы и Азии;
- » автоматизация ресторанов, парков, гостиниц (HelpLine);
- » маркетинг и продвижение мобильных приложений, игр (Unicume).

На площадке Speed Dealing встретились 25 представителей стартапов и 5 — венчурных и инвестиционных компаний.

Самыми перспективными инвесторы назвали:

- » Артура Манукяна (проект RoundCover — поставщик услуг расширенной гарантии для гаджетов и бытовой техники);
- » Дмитрия Иоффе (мобильное приложение для вузов, просмотр учебного расписание группы, преподавателей и друзей из других учебных заведений);
- » Анатолия Конухова (проект Eltechs — уникальная кросс-архитектурная виртуализация, позволяющая сочетать с новыми ARM-серверами ранее несовместимое ПО).

На стенде MATE CAREER собрались ведущие работодатели страны:

- » HeadHunter — крупнейший в Рунете сайт по построению карьеры для квалифицированных специалистов и менеджеров;
- » HR-Vision — профессиональный подбор специалистов выше среднего звена для российских и зарубежных компаний;
- » ITmozg — единственный в России спецресурс по поиску IT-сотрудников;
- » Ассоциация консультантов по персоналу (АКПП) — самое авторитетное профобъединение рекрутеров в РФ и странах СНГ.

Сотрудников искали и крупные компании-разработчики — Navitel и Appreal.

MATE оказалась хорошей платформой для обмена и демонстрации новых идей. Подробную информацию о Mobile Application & Technology Expo, фото и видео с выставки и конференции, а также презентации докладов см. на официальном сайте: <http://www.mate-expo.ru/>.

ДИЗАЙН ИМЕЕТ ЗНАЧЕНИЕ

Секреты разработки для Android

Не все стандарты и рекомендации соблюдаются...

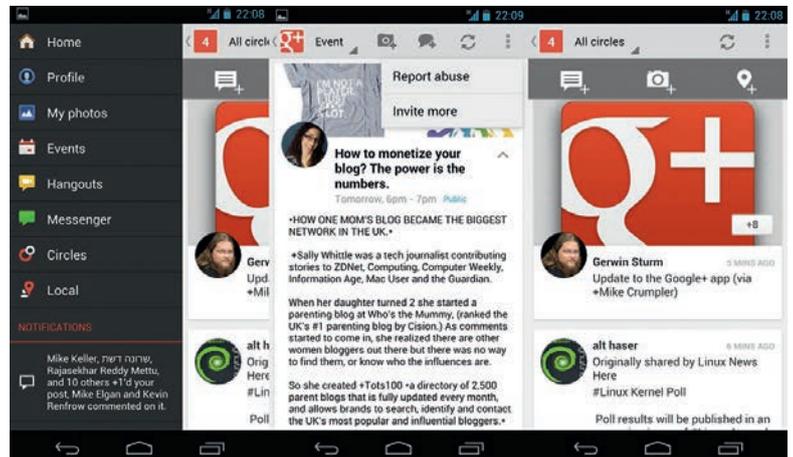
Google впервые опубликовал свои рекомендации (guidelines — «гайдлайны», если следовать жаргону разработчиков) для разработчиков еще при анонсе Android Honeycomb. Но тогда это осталось почти незамеченным: по пальцам можно пересчитать приложения, которые были оптимизированы для планшетов (а эту цель и преследовал Google). Вместе с презентацией 4-й версии своей ОС в Google сделали уже больший акцент на наведении порядка в магазине приложений Google Play, и появилось понятие *holo*-дизайна. Ранее лишь несколько программ следовали его принципам.

Дамир Франк [Damir Franc], эксперт по разработке мобильных приложений из Индии, отметил, что Google стоит начать с самой себя, исправив свои же приложения. Также, по его мнению, когда крупная компания создает некие правила, то все думают, что она будет их соблюдать. Microsoft реализовал все встроенные в Windows Phone программы как образец Metro UI, Apple до недавнего времени всегда использовал для оформления своих приложений т.н. скевоморфизм (*англ.* Skeuomorph от *греч.* Σκεῦος — «сосуд», «орудие», μορφή — «форма») — элемент дизайна, скопированный с формы другого объекта, но изготовленный из других материалов или иными методами (например, компьютерный календарь, отображающий дни как бумажный настольный). А Google умудрился в своих двух наиболее значимых (для себя) и продвигаемых программах по-разному реализовать боковую панель: речь идет о YouTube и Google+.

Добраться до sidebar'a в приложении YouTube можно, просто потянув в любой части экрана вправо: интуитивно, быстро и красиво. С Google+ сложнее: надо попасть в маленькие иконки в верхней части дисплея; диагональ наиболее популярных Android-смартфонов превышает 4,5 дюйма; совокупность этих факторов сильно ухудшает опыт пользователя от приложения в частности и телефона в целом.

В самом приложении слайды от края до края не используются, поэтому остается непонятным, почему боковые панели открываются лишь по нажатию на мелкую кнопку. Здесь уместен пример Gmail, который использует сразу два вида жестов: по элементу (письмо идет в архив) и от края (открывается боковая панель). Чего ждать от «вольняшек», если Google

Google+ пока остается эталонным «родным» приложением для Android на основе новых рекомендаций Google.



сам не приводит свои приложения к одному знаменателю?

Да, первые гайдлайны были выпущены еще во времена Honeycomb, но бума обновлений не произошло. Twitter лишь в марте 2013 года попытался привести свой клиент в соответствие с идеями Holo, Skype тоже ввел поддержку планшетов не так давно. Обновление дизайна в каждой крупной программе воспринимается как праздник. Новые приложения от разработчиков-одиночек, такие как Robird, соответствуют рекомендациям, работают быстро и хорошо. Безусловно, таким проектам проще проводить радикальные изменения, при отсутствии механизма проверки и согласований, но прошло уже больше 2 лет — может, уже пора и Google взяться за себя, замечает Дамир Франк.

«Проще проводить изменения при отсутствии согласований.»

Google заинтересован в разработчиках стороннего ПО для своей системы — что доказывает появление Android Studio. Не все посевы дают быстрые плоды... если вообще дают, как с Honeycomb.

Разработчику надо добавить всего два элемента, чтобы взаимодействие с его приложением стало приятнее: функцию «pull to refresh» и Action Bar. В первом случае у пользователя есть два варианта: использовать такой метод обновления или нет. Стандартный Facebook и абсолютное большинство Twitter-клиентов обладают такой возможностью, а вот встроенный

Google+ опять выступает примером, как делать не надо. Gmail получил эту функцию лишь с последним обновлением.

Когда Google заявили об отказе от кнопки меню, на авансцену вышел Action Bar, а вместе с ним — Nexus с диагональю экрана в 4,65 дюйма. А еще Xperia S, One X, Galaxy S3. Теперь в моде 5 дюймов, а большинство приложений продолжает размещать Action Bar сверху. А без кнопки меню это крайне неудобно. Лишь недавно такие активисты *holo*-дизайна, как Robird и Falcon Pro, перенесли управляющие элементы экрана вниз, где с ними удобно взаимодействовать. Play Books, Google Play, YouTube, Google+, теперь еще и Gmail — во всех этих приложениях Action Bar расположен сверху.

Сторонние же программы обладают странным свойством: если в телефонной версии приложения Action Bar расположен внизу, то в планшетной — все управляющие элементы вверх. Примеры — упомянутый выше Robird и почтовый клиент K10. Удачно сделан Action Bar в Falcon Pro: пользователь сам определяет положение управляющих элементов. Другим разработчикам стоит это тоже внедрить.

Безусловно, Google создал основу для развития новых приложений: гайдлайны, Android Studio, бета-тесты — но не создал мотивации. В Google Play наконец появились красивые приложения, но пока они скорее исключение; да и нужно дожидаться, когда старые приложения перейдут новый дизайн. Если разработчики будут правильно использовать такие вещи, как Action Bar, pull-to-refresh, боковую панель и другие, то качественных программ станет гораздо больше. LXF

Сравнение

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

Дистрибутивы для планшетников x86

Хотите запустить Linux на своем ослепительно-новом планшете? Дэвид Хейвард тестирует пять подходящих дистрибутивов.



Про наш тест...

Нам в руки попал отличный Acer Iconia W500, с двоядерным 1-ГГц AMD-C50 CPU, 2-ГБ ОЗУ DDR3, 10,1-дюймовым экраном WXGA 1280×800, AMD Radeon GMA 6250 и 32-ГБ SSD. Есть также Bluetooth, Wi-Fi, две камеры (спереди и сзади), порты USB и MicroSD — в общем, очень неплохой набор.

Кроме того, в нем имеется функция акселерометра, которая вдобавок позволяет планшету переворачивать экран, если повернуть его на бок.

Нам нужен был дистрибутив, отвечающий всем потребностям всех функций, которые должен предлагать планшет, желательно прямо сразу после установки, поэтому мы выбрали пять последних релизов — Ubuntu, Android x86, Fedora, Kubuntu Active и openSUSE — и проверили их возможности в виде live USB или после установки, чтобы посмотреть, далеко ли мы можем зайти в этой области.

Наша подборка

- » Android x86
- » Fedora
- » Kubuntu Active
- » OpenSUSE
- » Ubuntu

Мы живем в мире, где бал правят планшетники или смартфоны — и в жизни, и онлайн. Вместо того, чтобы купить на Рождество новенький ноутбук, теперь предпочитают один из множества 10-дюймовых планшетов, которые выставлены повсеместно.

И это забавно: 15 лет назад единственным планшетником был DataPad капитана Звездного флота Жана-Люка Пикара [Jean-Luc Picard]. Никогда старая аксиома, что бывшая научная фантастика становится научным фактом, не была столь правди-

ва. Мы прожили уже почти пол-2013 года, а рост и количества, и популярности планшетников продолжается, причем независимо от того, Android это или Apple. Так что, как говорится, если не получается победить, присоединяйтесь. Нам в LXF пришла в голову умная мысль (вот лишнее доказательство, что такое бывает и с нами):

а почему бы не взять планшетник x86 и не установить на нем Linux? В конце концов, если это устройство x86, то мы ведь можем установить на него практически все, что угодно, верно? Ну и, плюс к тому, что ни возьми, хуже, чем Windows 8, не будет. Так что получайте наше Сравнение дистрибутивов для планшетников.

«15 лет назад единственным планшетником был DataPad капитана Жана-Люка Пикара.»

Установка

Все ли работает на самом деле?

Мы почти полностью уверены, что нам незначительно рассказывать, как установить дистрибутив Linux — во многих сегодня используется понятная и несложная программа установки; однако сочли, что было бы интересно посмотреть, как они справятся с работой — сначала в качестве образа live, через загружаемый USB, а затем в качестве полноценно установленной ОС.

Ubuntu не нуждается в представлениях. Это самый распространенный дистрибутив Linux, и вы, вероятно, уже знаете, что Canonical выпускает Ubuntu для смартфонов, так что последняя редакция этой настольной ОС выглядит самым подходящим началом. Загрузка в сессию live прошла успешно, так же, как и установка на SSD Iconia. Wi-Fi, Bluetooth, обе камеры, сенсорный экран и звук заработали сразу. К сожалению, не было доступа к экранной клавиатуре, когда мы затеяли ввод в текстовое поле, но мы решили эту проблему с помощью настроек Onboard (которые открываются из Dash), и клавиатура появилась, как только мы начали печатать.

Порт x86 Android не захотел устанавливаться (он постоянно перезагружал Iconia), однако сессия live прошла отлично. Все работало, в том числе и поворот экрана, однако экранная клавиатура отсутствовала, и нам пришлось подключить беспроводную. Однако уже через минуту мы смогли войти и начать играть.

Kubuntu Active был первой средой Plasma Active, и хотя выглядел впечатляюще, все же в среде live мало что работало. Установка прошла вполне ровно — все работало, за исключением поворота экрана — однако спустя примерно минуту все застопорилось. То же произошло и после перезагрузки.

Fedora 18 работает на Gnome 3.6.3, который отчасти позволяет тачскрину работать лучше, чем в более ранних версиях Gnome. Горячий угол [hot corner] легко активируется, скроллинг такой же качественный, как и все прочее, предлагаемое Android, и значки программ достаточно велики по размеру, чтобы запускать именно то, что нужно, не задевая соседних значков. Звук, Wi-Fi, Bluetooth и камеры заработали сразу, и мы смогли активировать



» OpenSUSE обладает наилучшей функциональностью на планшете, если применить модель установки TabletPC.

экранную клавиатуру, прикоснувшись к значку Accessibility в верхней части рабочего стола (она также появляется, когда вы прикасаетесь к текстовому полю).

Сессия live openSUSE 12.2 KDE работала идеально; все оборудование, как и обычно, заработало сразу. Сенсорный экран, однако, хоть и отзывался, но не поворачивался; зато когда мы выбрали модель установки TabletPC после установки на SSD из YaST, перед нами открылся новый уровень удобства использования Linux. Заработали и мультисенсорная панель, и разворот экрана, и при касании текстового поля запускалась экранная клавиатура по умолчанию, и мы смогли использовать планшет так же, как любую версию Android или Windows 8.

Вердикт

- OpenSUSE ★★★★★
- Fedora ★★★★★
- Ubuntu ★★★★★
- Android x86 ★★★★★
- Kubuntu Active ★★★★★

» OpenSUSE стал победителем — с установкой по модели TabletPC.

Программы по умолчанию

Что вы получите сразу после установки?

В этом тесте выносить оценку было довольно трудно — полные дистрибутивы рабочего стола создают полноценные пакеты программ по умолчанию; но разница все же есть. И снова нам не приходится много говорить об Ubuntu:

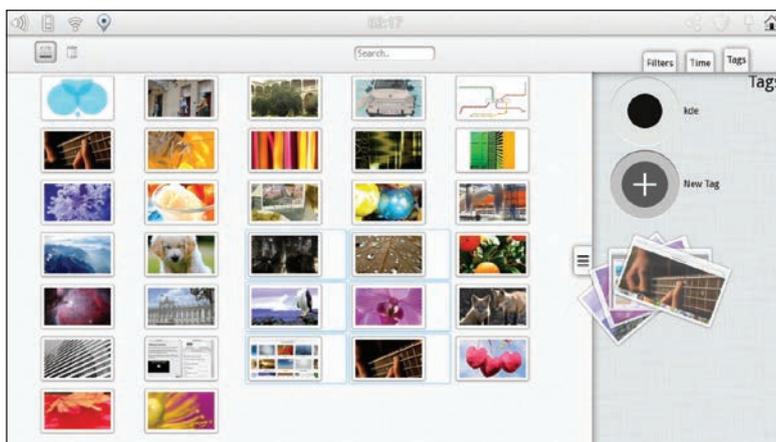
вы получаете то, что получаете, и мы все знаем, что в нем есть.

Больше всего нас удивил Kubuntu: мы полагали, что образ Kubuntu Active не будет бловать разнообразием, концентрируясь на Plasma Active, и были приятно

удивлены тем, что это оказалось не так. Kubuntu предлагает *Calligra Suite*, а также *Krita* и *Kexi* для работы с изображениями и базами данных.

Что касается программ по умолчанию Android x86, картина здесь оказалась куда менее впечатляющей. Несколько игр, видеоплеер, менеджер изображений; но если не считать обычных настроек, карт, Gmail и т.д., он мало чем отличается от тестовой среды разработки.

Как и следовало ожидать, Fedora и openSUSE так же хорошо упакованы, как и Ubuntu, что делает планшет более удобным устройством сразу после установки. Однако в Fedora мы наткнулись на неприятную проблему: когда бы мы ни пытались открыть *Writer*, он почти сразу же выдавал отказ, так же, как и *Impress*. Мы переустанавливали его несколько раз, но это не помогло. Возможно, проблема скрывалась именно в нашей установке, поскольку в среде Live USB оба инструмента работали прекрасно.



» Kubuntu Active предлагает намного больше программ, чем мы ожидали.

Вердикт

- Kubuntu Active ★★★★★
- OpenSUSE ★★★★★
- Ubuntu ★★★★★
- Fedora ★★★★★
- Android x86 ★★★★★

» Все они — полнофункциональные дистрибутивы рабочего стола, но скрпизмом стал Kubuntu Active.

Внешний вид и удобство

Взглянем правде в глаза: ну млеем мы от сексуального вида планшета...

Планшеты выглядят холеными, блестящими, в них много лоска, отличная анимация и привлекательные значки. Работают они или нет — это уже другой вопрос, но выглядят-то они шикарно.

Установив дистрибутив рабочего стола на наш планшет x86, мы заполучили все, что только возможно — отличный вид, гаджеты и виджеты,

которые можно найти на любом превосходном ПК, но всем этим прелестям тем не менее нужно работать на планшете, чья мощность отнюдь не настолько велика.

Как и в большинстве систем, где графические прелести перевешивают функциональность, приходится чем-то жертвовать. Пускай нам и хотелось, чтобы наши дистрибутивы выглядели

отлично, но мы также не собирались перегружать ресурсы планшета до того, чтобы он в конечном итоге оказался бесполезным.

Однако, приложив некоторые усилия, мы умудрились достичь известного равновесия и предоставить каждому из пяти дистрибутивов Сравнения возможность честно отработать затраченные на них деньги.

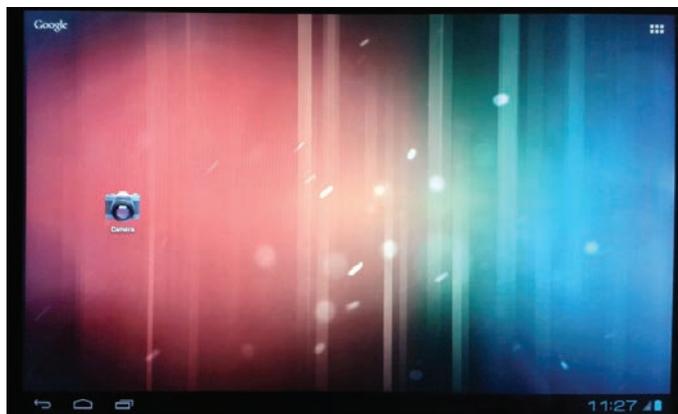
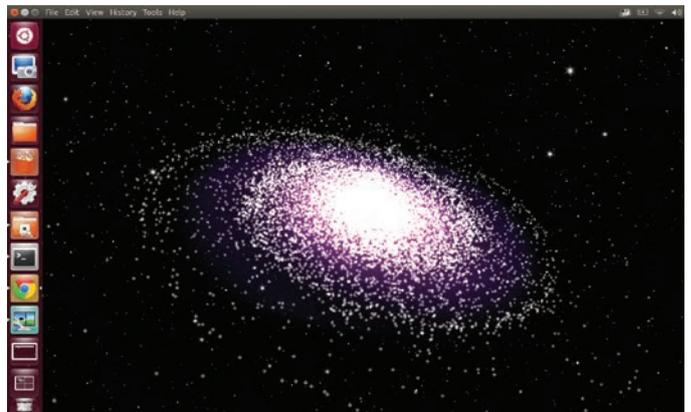
Ubuntu ★★★★★

От Unity многие пользователи по-прежнему ворочают нос, не так ли? И, между прочим, это не совсем правильно.

Мы добавили LiveWallpaper с launchpad.net/livewallpaper и наслаждались эффектом спиральной галактики на нашем планшете. Рейтинг нашего внешнего вида резко поднялся, но система еле ползала: все-таки графические эффекты — штука накладная.

Можно установить и другие среды, но следует отдать дань справедливости Unity: несмотря на общую враждебность к этому рабочему столу, он работал действительно хорошо и выглядел тоже неплохо.

Удобство Ubuntu, и, следовательно, Unity, на планшетном компьютере неоспоримо. С помощью мелких настроек вы можете добиться вывода экранной клавиатуры, и вам ничто не мешает сделать Ubuntu своей постоянной ОС на планшете x86.



Android x86 ★★★★★

Android x86 не так широко известен, как ARM-версия, что вполне очевидно — в конце концов, это всего лишь портирование. И прежде чем вступать в дебаты на тему «Android — это не Linux», стоит упомянуть, что в данном примере мы рассматриваем его именно в таком качестве, и поскольку это самая популярная ОС для планшетов, мы бы манкировали своими обязанностями, если бы не включили ее.

По части внешнего вида Android прекрасно справляется со своей задачей. Родные обои просто выдающиеся, значки для приложений и виджетов выглядят отлично, но мы не смогли заставить работать обои live — они подвели планшет намертво. Количество приложений в Android огромно — для ARM-версии. Для нашей версии, проекта x86, приложений негусто; то, что имеется, обладает скверной привычкой заставлять все остальное сбоить, однако основные установленные приложения работали хорошо.

Функции планшета

Что с поворотом экрана и мобильной способностью к подключению?

Что делает планшеты столь популярными? Вопрос хороший. Не простота ли в использовании? Или — возможность просто взять их и выйти в Сеть? А может, огромное количество бесплатных приложений, доступных в наши дни? Мы думаем, что это в большей мере касается функций, предлагаемых планшетом: мобильное соединение, определение местоположения GPS и возможность повернуть устройство набок и применить как электронную книгу.

Поворот экрана в Ubuntu сначала был довольно проблематичным, но после того, как мы последовали инструкциям на goo.gl/pQWNJ, он заработал должным образом и заделался отличным eReader; то же самое было и с мультимедиа, и с соединением. При наличии Bluetooth, HDMI и двух портов USB было легко смотреть фильмы и играть в игры. То же можно сказать об openSUSE после установки TabletPC, поскольку именно она добавила функцию поворота экрана.

Kubuntu, к сожалению, оказался совсем другим. Отладка поворота экрана не работала, и по неизвестным причинам видео через HDMI было до такой степени искажено, что мы не смогли его разглядеть. В режиме планшета с экраном все было нормально.

Поворот экрана в Android x86 работал превосходно, но порты USB и HDMI — нет.

Поправка Fedora для поворота экрана просто-напросто убила всю установку. Но все остальное работало как положено.

Вердикт

- OpenSUSE ★★★★★
- Ubuntu ★★★★★
- Kubuntu Active ★★★★★
- Fedora ★★★★★
- Android x86 ★★★★★

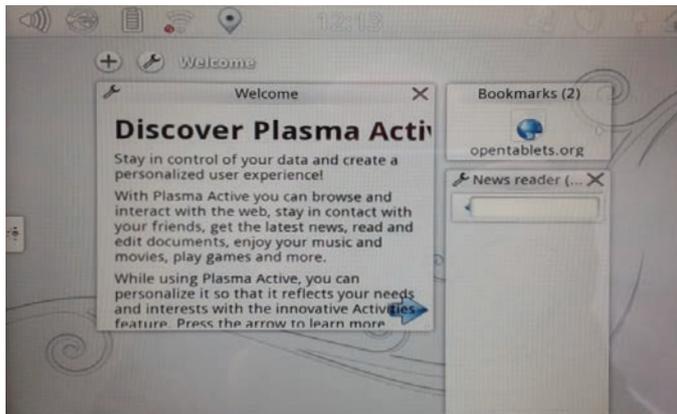
» OpenSUSE и Ubuntu лидируют — после применения поправок.

Fedora 18 Gnome ★★★★★

Использование Fedora Gnome 3.x вызвало почти такое же отторжение пользователей, как Unity, но его слабости на рабочем столе оказались сильными сторонами на планшете.

Gnome 3 выглядит очень хорошо — мы даже не стали заморачиваться такими забавными примочками, как обои live. Анимация (например, горячий угол) работает плавно и быстро, скроллинг выглядит отлично, да и значки симпатично анимированы. Если хотите попробовать его на другом дистрибутиве, имейте в виду, что Gnome 3 — очень приятная среда для планшетного компьютера.

Что до удобства в использовании, Fedora хороша настолько, насколько этого можно ожидать от дистрибутива рабочего стола со столь солидным стажем работы. Единственная проблема, с которой мы здесь столкнулись — ужасный процесс установки.



Kubuntu Active ★★★★★

Мы, кажется, уже говорили, как нас удивил Kubuntu Active и его среда Plasma Active. Для нашего Сравнения мы взяли Kubuntu Active 12.10, который использует — насколько нам известно — Plasma Active 2. Хотя в основном это совместный проект с командой KDE, его внешний вид сильно отличается. Он нацелен на использование на планшетах и смартфонах, и неудивительно, что он отлично выглядит, особенно с панелью Peek&Launch и Plasma Widgets.

Говоря об удобстве в использовании — как только вы проходите стадию начальной загрузки в среду live (которая запускается довольно вяло, но потом разгоняется до приличной скорости), и как только будет решена проблема остановок (мы установили новый образ), ОС раскрывается во всей красе, превращаясь в один из лучших примеров системы с сенсорным экраном, которую мы когда-либо использовали. Возможно, довольно смелое заявление, но вполне заслуженное.

OpenSUSE 12.2 KDE ★★★★★

Вы можете считать KDE в openSUSE бедным родственником среды Plasma Active — но только Active здесь новичок. Хотя он и не выглядит особо дружелюбным к планшетах, однако исправляется, когда дело доходит до удобства в использовании.

OpenSUSE 12.2 — превосходная настольная ОС, но при использовании на планшете и она, и KDE выглядят несколько архаичными, что особенно бросается в глаза после знакомства со средой Plasma Active из Kubuntu. Да, вы можете настроить openSUSE, и в весьма значительной степени, чтобы заставить ее выглядеть более «планшетно», но кнопки Start скоро окажутся среди вымирающих видов.

Ее удобство в использовании, однако, потрясающее, и, как мы упоминали в начале, установка по модели TabletPC возвращает систему к современной жизни.



Gnome vs KDE vs Unity

Какая среда рабочего стола лучше всего работает на планшете?

Различные среды рабочего стола на ПК работают так, как и должны, даже если нам они не нравятся. Однако на планшете мы используем разные движения пальцев для навигации на рабочем столе, и если это не работает должным образом, то даже выполнение самых простых задач становится раздражающим.

Хотя Unity и ориентирована на планшет, мы очень быстро разочаровались, поняв, сколько трудов уходит на то, чтобы просто закрыть, минимизировать или изменить

размер открытых окон. Unity создана для этого — частично — когда проведение пальцем сбоку выводит Unity Launch Bar.

Переключение между приложениями в Android x86 не было проблемой, но зато он постоянно принимался чудить: одни приложения случайным образом закрывались, а другие запускались явно по собственному усмотрению.

Работа с Fedora и Kubuntu была истинным наслаждением: все движения пальцами делали Gnome 3 и Plasma Active

лучшими средами рабочего стола в нашем тесте, хотя и без поворота экрана.

Бедный старина openSUSE оказался не на высоте, когда дошло до употребления кнопки Start, но если не полениться создать значки для приложений, то потребуется всего лишь касание экрана. Стоит еще раз отметить, что openSUSE стремительно хорошеет при установке TabletPC. Единжды установив его, вы получите один из самых лучших рабочих столов для использования на планшете.

Вердикт

- Fedora ★★★★★
- Kubuntu Active ★★★★★
- OpenSUSE ★★★★★
- Ubuntu ★★★★★
- Android x86 ★★★★★

» В этой номинации победу поделили Gnome 3 и Plasma Active.

Ежедневная работа

Как дистрибутивы справятся с повседневными задачами?

Тестирование разных дистрибутивов на планшете x86 — истинное удовольствие. Однако, работая над этим проектом, мы использовали каждый дистрибутив кратковременно, а потом переходили к следующему. Ну, а как насчет долгосрочного использования? Какой из пяти наших дистрибутивов будет лучшим для ежедневной работы?

Мы долго и упорно размышляли над этим разделом, потому что все зависит от того, что вы намерены делать со своим планшетом. В конечном итоге мы свели повседневное использование к следующему: проверка рабочей почты во время поездок;

создание документов; возможность поиграть в игру (ничего особо напряженного); возможность побродить по Интернету и зайти на Facebook/Twitter; видеозвонки; и возможность использовать его в качестве устройства для чтения eBook и воспроизведения мультимедиа.

Ubuntu начинает игру своим обычным набором программ по умолчанию. *Firefox*, *LibreOffice* и игры через Software Centre работали, как положено, в основном предоставив нам все, что нужно. Для видеозвонков мы установили *Skype 4.1* для Linux, который работал весьма достойно, а также *FBReader* — читать eBooks (когда наладили

поворот экрана) и *VLC* и разные кодеки для воспроизведения мультимедиа.

Fedora и Kubuntu работали с базовыми программами, но в Kubuntu *Skype* постоянно слетал, и когда дело дошло до чтения электронных книг, возникла проблема с поворотом экрана.

Android x86 оказался довольно бесполезен в качестве ОС для планшета для повседневной работы. В нем нет рабочих текстовых редакторов, а web-браузер вообще не запускался. С играми все было нормально, но мультимедиа тоже отказались работать, и, опять же, никакой связи по видео. Что бы мы ни делали, оно либо не работало, либо тормозило планшет.

OpenSUSE оказался самым удобным. Все работало превосходно: *LibreOffice*, *Firefox* и несколько игр были на высоте. *Skype* тоже легко добавлялся через RPM со Skype.com, и опять же мы использовали *FBReader* для openSUSE. Дополнительные прелести, получаемые благодаря TabletPC — это единственная причина, по которой openSUSE в данном разделе превосходит Ubuntu: жесты действительно помогают ОС работать как полагается. Более того, хотя openSUSE загружается чуть дольше, чем Ubuntu, он быстрее делал привязку к точкам Wi-Fi, быстрее выходил из спящего режима и дольше держал заряд, что делает его идеальным выбором для долгосрочного использования.



➤ Большинство дистрибутивов предлагают набор основных инструментов для повседневного использования — кроме Android x86, в котором нет даже текстового редактора.

Вердикт

- OpenSUSE ★★★★★
- Ubuntu ★★★★★
- Fedora ★★★★★
- Kubuntu Active ★★★★★
- Android x86 ★★★★★

» OpenSUSE просто оказался лучшей ОС для долгосрочного использования на нашем планшете.

Будущее

Где наш хрустальный шар?..

Сanonical надеется через несколько месяцев захватить мобильный мир — и судя по тому, что мы видим в Ubuntu Touch, их мечты вполне могут превратиться в явь.

И остальные дистрибутивы берут с них пример, разрабатывая свои самые свежие или готовящиеся к выходу версии для мобильных устройств. Итак, что же в запасе у разработчиков для наших пяти дистрибутивов?

Проект Android x86 набирает силу, однако им еще придется немало поработать, чтобы добиться полной интеграции с ARM-версиями.

Fedora 19 (Schroedinger's Cat — кошка Шредингера) обещает предоставить улучшенный способ работы на планшете в виде

большой поддержки мультисенсорного экрана в ядре и лучшего дизайна пакетов для пользователей планшетов — так что, возможно, стоит подождать выхода Fedora 19, а не начинать с Fedora 18.

Plasma Active в Kubuntu неуклонно движется вперед. На горизонте вырисовывается интенсификация использования OpenGL, а Plasma Active 5, основанная на KDE Frameworks 5, обещает стать событием года. На самом деле, 2013 год весьма благоприятен для Plasma Active, хотя еще многое предстоит сделать.

OpenSUSE находится в активной разработке, однако специальных планов по поводу версии для планшета у него нет. Очевидно, что превосходный TabletPC будет разрабатываться и в дальнейшем,

но не будет интегрирован в основную ОС. Мы не уверены в том, что произойдет с этим дистрибутивом в будущем — а те, кто точно знают это, хранят свои знания в секрете.



➤ Возможности, предоставляемые Ubuntu Touch, заставляют полагать, что именно дистрибутив от Canonial — отличная заявка на долгосрочное использование.

Вердикт

- Ubuntu ★★★★★
- Fedora ★★★★★
- Android x86 ★★★★★
- Kubuntu Active ★★★★★
- OpenSUSE ★★★★★

» Представьте себе мир, где правит Ubuntu — это вполне реально...

Дистрибутивы для планшетников x86

Вердикт



Работая над нашим Сравнением, мы достаточно глубоко погрузились в мир планшетов x86. Они все больше и больше производят впечатление бедных родственников более успешных ARM-версий. Очень жаль, поскольку потенциал планшета x86, на наш взгляд, куда выше, чем у ARM. Скажем для новичков, что у программ для x86 долгая история — различные отладки и настройки, дополнения и добавки, которые персонализировали наши рабочие столы Linux.

Планшет x86 просто имеет ряд преимуществ. Представьте себе, что вы установили Linux Steam Client на довольно мощный планшет i3/5 x 86 и начинаете погружаться в мир достойных игр для Linux. И еще представьте себе свой любимый дистрибутив, дополненный персонализированными украшениями окон, тщательно настроенной *Conky* и обоями DeviantArt. Планшет x86, без всяких сомнений, превосходит для этого подойдет.

Итак, куда это все ведет с нашей подборкой дистрибутивов для x86 планшета? В лидеры вырвался Ubuntu — однако буквально на пятки ему наступает openSUSE. Kubuntu Plasma Active и Fedora 18 Gnome заняли третье и четвертое место, а проект Android x86 попал на пятое.

Топ-планшеты

Как и следовало ожидать, у всех дистрибутивов есть свои сильные и слабые стороны. Ubuntu явно лидирует в марше планшетов со своими планами на будущее, но пока это ОС, в первую очередь ориентированная на настольное применение.

openSUSE идет вторым, в основном благодаря впечатляющему TabletPC — без него не видать бы ему лавров. Kubuntu Plasma Active — что очевидно — разработана для планшетов, и было приятно сравнить обычный KDE в openSUSE с Plasma Active KDE в Kubuntu; но над обоими стоит потрудиться, чтобы они хотя бы приблизились

к возможности стать эквивалентами Android на планшетах ARM и смартфонах.

Fedora склоняется в сторону планшетов, и Gnome 3 стал симпатичным элементом рабочего стола планшета. Но стоит внимательно присмотреться к Fedora 19 и выше — мы полагаем, что в будущем Fedora получит большее распространение на планшетах, чем любой другой дистрибутив.

И последнее место — хотя далеко не по значимости — занимает дистрибутив Android x86. Вы, конечно, можете сказать, что это лишь любительский проект, но потенциал его огромен, и при хорошей поддержке и разработке Android вполне сможет штурмом захватить мир x86.

» Ubuntu возглавляет на себя корону дистрибутива для планшета x86, но openSUSE становится вторым с очень небольшим отрывом.

I Ubuntu 12.10 ★★★★★
 Сайт: goo.gl/0wKz9 Лицензия: Свободная
 » Хорошо выглядит и хорошо работает. Кто знает, что будет дальше...

IV Kubuntu Active ★★★★★
 Сайт: goo.gl/MjnNt Лицензия: Свободная
 » Нужна доработка, но Plasma Active 5 намекает на замечательные перспективы.

II OpenSUSE 12.2 KDE ★★★★★
 Сайт: goo.gl/2MibT Лицензия: Свободная
 » OpenSUSE с TabletPC занимает второе место с очень небольшим отрывом.

V Android x86 ★★★★★
 Сайт: goo.gl/24DKx Лицензия: Свободная
 » Требуется немало работы, чтобы сделать его достойным родственника для устройств ARM.

III Fedora 18 Gnome ★★★★★
 Сайт: goo.gl/FiJH Лицензия: Свободная
 » Fedora 19 могла бы стать главной ОС для планшета x86. Поживем — увидим.

Обратная связь
 Вы согласны? Или хотите попробовать создать дистрибутив для планшета x86? Напишите нам на lx.f.letters@futurenet.co.uk.

Рассмотрите также...

Есть множество разных дистрибутивов, которые мы могли бы установить и повозиться с ними, чтобы получить полное впечатление о ситуации с x86 планшетом, однако нам действительно хотелось начать с самых простых и отнимающих меньше всего времени опций. Если выбранные нами дистрибутивы вас не радуют, мы бы посоветовали вам обратить внимание на Zorin 6.2.

Zorin — многофункциональный дистрибутив, разработанный специально для пользователей Windows, которые хотят легко и плавно переехать в Linux; и к тому же симпатичный, быстрый и довольно гибкий. Основанный на Ubuntu, он со временем определенно похорошел, и на нашем планшете просто блистал. Он быстро загружается, отзывчив к касаниям и жестам и предоставляет

терпимую экранную клавиатуру — хотя мы в итоге заменили ее на *Maliit*, которая выглядела лучше и работала эффективнее.

Итак, если ваш проект планшета x86 на Linux забрел не туда, куда вам хотелось бы, загляните на страницу загрузки Zorin, zorin-os.com/free.html, и дайте ему шанс. Мы уверены, что вы не разочаруетесь. **LXF**



Что утратил Ubuntu?

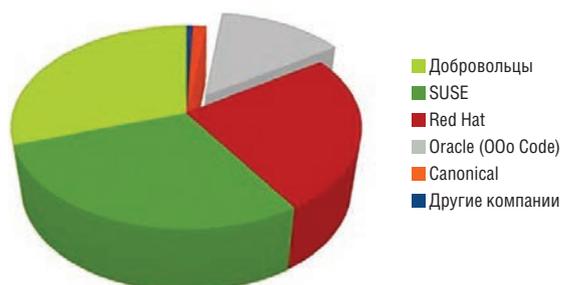
Не утратило ли сообщество Ubuntu свои права
из-за стремления Canonical к конвергенции платформ?
Сюзан Линтон выяснит это...

Когда почти десять лет назад Ubuntu вырвался на сцену, он самым своим названием четко обозначил свою позицию. Слово «ubuntu» означает «мы — те, кто мы есть ради других». Принимая этот моральный принцип в качестве основополагающего, следовало ожидать, что дистрибутив Ubuntu сконцентрируется на разработке сообщества, разделении влияния и попытках привить чувство сопричастности своим пользователям и разработчикам. Философия сообщества предполагала, что каждый, кто принимает участие в жизни дистрибутива, имеет долю в обладании им.

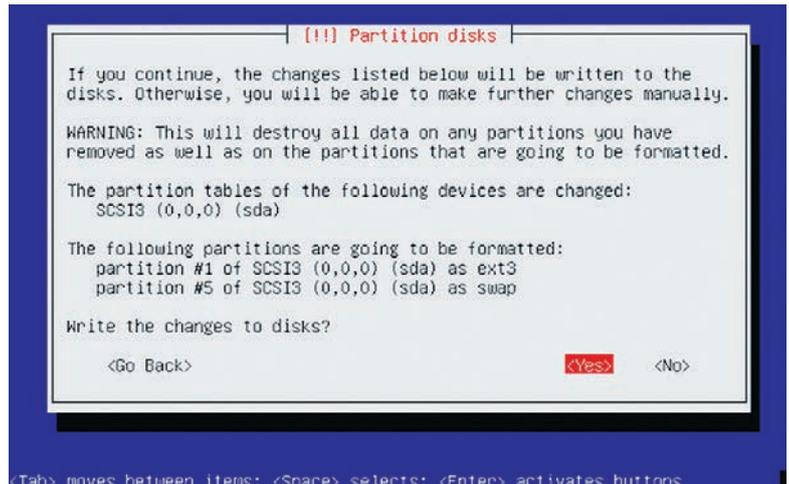
С самого начала Ubuntu концентрировался на сообществе. На роль менеджера сообщества Ubuntu был назначен Бенджамин Мако Хилл [Benjamin Mako Hill], являвшийся ранее активным участником сообщества Debian. Затем эта роль перешла к Джоно Бэкону [Jono Bacon], который регулярно пишет статьи для LXF и весьма серьезно относится к тому, чтобы объединить людей во имя создания чего-то особенного. Среди проектов свободного ПО почему-то имеется тенденция слушать того, кто громче всех кричит. Ubuntu с самого начала старался сделать так, чтобы было слышно всех. Если форумы других дистрибутивов бывают не очень-то приветливы к новичкам, которым просто дают совет «RTFM (read the f***** manual [читайте ... руководство пользователя])», то форумы Ubuntu всячески поощряют всех и каждого делиться своими знаниями и опытом с другими, а не просто выставлять таковые напоказ.

Это кажется идеализмом, утопией, и пусть так оно и было, но некоторые ощущали, что хоть все и равны, но кое-кто из участников все же равнее. Одна из жалоб, с самых верхов, принадлежит Грегу Кроа-Хартману [Greg Kroah-Hartman], разработчику ядра для стабильной ветви, который выявил несоразмерность работы, выполняемой Canonical, и работы, выполняемой другими дистрибутивами. В своем выступлении на Linux Plumbers Convention в 2008 г. он подверг Ubuntu критике за то, что там больше не вносят своего вклада в разработку ядра Linux, заявив, что из 100 000 заплат ядра, созданных в течение последних пяти лет, только 100 были присланы из Canonical, что создает странную ситуацию, когда доля самого популярного в мире дистрибутива Linux составляет 0,1 % от всей работы, нужной для обеспечения работы ядра. За тот же период доля Red Hat составила 11,9 %, а доля Novell, создателей SUSE, 7,3 %.

Недавние цифры, полученные после выхода ядра Linux 2.6.36, обнаружили похожую статистику. Red Hat, SUSE, Samsung (благодаря своей работе над Android), IBM, Google, AMD, Nokia и даже Microsoft — все внесли свой вклад в работу над ядром, и он оказался больше, чем вклад Ubuntu. Четыре года спустя, в 2012, Кроа-Хартман пожаловался, что «Canonical употребляет меня как этакого привратника, который следит, чтобы в пакете ядра были исправлены ошибки, а потом отправлены клиентам. Это настолько неправильно, что у меня просто руки опускаются...»



» Ubuntu вносит свой вклад в другие проекты — просто этот вклад не так велик, как у SUSE, Red Hat, Oracle...



» Все та же «трудная» программа установки Debian чудесным образом превратилась в «простую», попав в Ubuntu — стараниями множества писателей и блоггеров.

Launchpad

В 2004, вскоре после запуска Ubuntu, Canonical объявила о появлении платформы разработки для размещения программных проектов, которая включила множество важных для разработчиков функций, собранных на привлекательном и удобном портале. Ubuntu должен был стать всего лишь одним из тысяч проектов, которые должны были храниться, обновляться, отслеживаться, планироваться и управляться через Launchpad. У остальных сайтов, размещающих проекты, были свои проблемы, и Launchpad был создан, чтобы решить их.

Однако, несмотря на присягу моральному принципу открытости, Canonical в течение пяти лет держала свой код web-разработки закрытым, и переход на Affero GPL совершился только в июле 2009 (лицензия Affero закрывает просчет стандартной GPL — когда код запускается удаленно через web-сервер, он на самом деле не распространяется, и поэтому GPL неприменима. Affero решает эту проблему, чтобы web-приложения оставались открытыми). Спустя четыре года — и шквал критики — после того, как сервис был впервые открыт, Launchpad стал программой с открытым кодом, однако осталось впечатление, что Ubuntu в теории провозглашает одно, а делает на практике совсем другое.

Что утратил Ubuntu-метр



» Местные организации и сообщества должны получить одобрение (или неодобрение) Canonical на продвижение Ubuntu. Бесплатные DVD по-прежнему им выдаются.

Споры рабочего стола

Ubuntu фундаментально меняет дизайн, но так ли это нужно?

Ubuntu добился популярности с помощью рабочего стола Gnome. Вливание в Debian подборки самых передовых пакетов и предсказуемый цикл релизов плюс самый свежий стабильный релиз рабочего стола Gnome — все это шло на пользу не только Ubuntu, но и Gnome.

В 2004 году, когда вышел Ubuntu, Member's Choice Awards [Награды по выбору участников] на LinuxQuestions.org провозгласили KDE самым популярным рабочим столом: он получил 58,25 % голосов в открытом голосовании, в то время как 30,9 % были отданы за Gnome (третье место по популярности занял Xfce, набрав 11,23 %).

По мере роста известности Ubuntu росла и популярность Gnome. В 2008 году разрыв сузился: 43 % в пользу KDE и 40 % в пользу Gnome. К 2010 году (через два года после того, как KDE сломал свою парадигму версией 4.x) Gnome превзошел по популярности KDE, набрав 45 % голосов как лучшая среда рабочего стола, тогда как KDE набрал 33 %. Будущее было блестящим. Оно виделось в розовом цвете. Пользователи Ubuntu полюбили Gnome.

Навязанные изменения

Стало понятно, что происходит нечто неладное, когда пользователи, установившие обновления до версии 10.04, перезагрузились и обнаружили, что кнопки в окне мистически перебрались с правой стороны панели окна на левую. Отвечая на полученные сообщения, основатель Canonical Марк Шаттлворт [Mark Shuttleworth] сказал, что «перенос влево прекрасно освобождает пространство справа, и в версии 10.10 мне бы хотелось поэкспериментировать с некоторыми новаторскими опциями. Команда разработки отлично осознает всю спорность ситуации, и ваши (желательно вежливые) комментарии и, что еще важнее, данные очень приветствуются и помогут нам принять наилучшее решение».

Иванка Маич [Ivanka Majic] из команды разработчиков Ubuntu объяснила это решение как «золотой шанс сделать нашу ОС не только конкурентоспособнее, но и лучше», и спросила: «Мы что, обкурились, чтобы считать трудность привыкания к новому положению достойной отката от всех инноваций в принципе?»

К сожалению, освоение и привыкание оказались для большинства пользователей слишком трудными, и они предпочли вернуть кнопки на их законное место с помощью *gconf-edit*. По сию пору Google на запрос «кнопки окна справа» выдает руководство по исправлению Ubuntu —

и это из-за изменения дизайна, имевшего место два с половиной года назад!

В противоположность принципам открытости, о которых Ubuntu заявлял раньше, Шаттл-

ворт объяснил отношения Ubuntu с большинством пользователей, сообщив в списке рассылки: «У нас есть процессы проверки, что мы хорошо умеем делегировать задачи, однако не стоит считать, что открытое сообщество означает право голоса каждого во всем. Это не демократия. Хорошая обратная связь, достойные данные приветствуются. Но мы не проводим голосования по инженерным решениям».

Спустя два с половиной года после начала эксперимента со строкой заглавия мы все еще ждем тех инноваций на правой стороне панели. Зато уже получили исток всех инцидентов по выплескиванию воды вместе с ребенком, в виде Unity.

Unity отнюдь не стал идеальным решением, как сулили в Canonical, и по сути стал достаточно стабильным для нормальной работы только в последних версиях. Одной из проблем с Unity был *Compiz*, на котором изначально и строился Unity и который частенько конфликтовал с OpenGL.

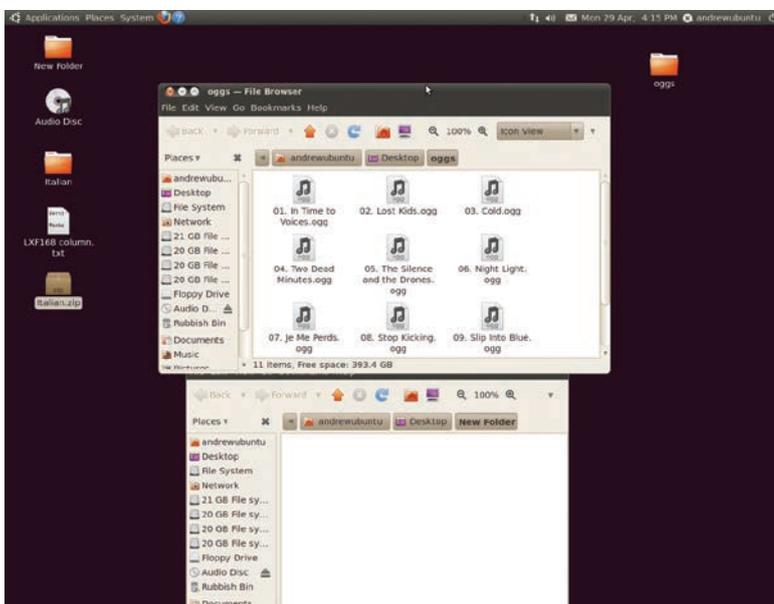
Будет справедливым сказать, что, если оставить в стороне технические проблемы, Unity не пользовалась популярностью. Если взглянуть на цифры за год после выхода Unity (2011) на LinuxQuestions.org, KDE по-прежнему остается любимым дистрибутивом для 33 % пользователей. Unity, который по идее должен был стать лучше Gnome, набрал всего 4,6 %. Возможно, более показательным стал тот факт, что популярность Xfce увеличилась до 27 %.

Весьма странно, что создав такое огромное и готовое помочь сообщество, Ubuntu перестал к нему прислушиваться. Изменения в дизайне — это только один пример, но есть и другие: демон *Upstart init* (который совершенно бессмысленно дублирует большую часть работы, выполняемой в SystemD) и Project Harmony — оба являют собой примеры того, как Ubuntu анонсировал новые идеи, которые как-то не очень пошли.

Другой идеей, не нашедшей воплощения, стало предложение перевести Ubuntu на модель разработки с возобновляемыми релизами либо параллельно, либо вместо обычного двухгодичного цикла. Дистрибутивы, использующие модель возобновляемых релизов, например, Arch или Manjaro, достаточно популярны среди опытных пользователей, и есть ряд других, которые либо являются исключительно возобновляемыми релизами, либо предлагают возобновляемую версию помимо периодических полных релизов. Однако те, кто занимается Ubuntu, обеспокоились.

«У нас есть процессы проверки умения делегировать задачи.»

Кнопки слева: да, это больше похоже на OS X, но в свое время это было крайне непопулярно среди пользователей Ubuntu.

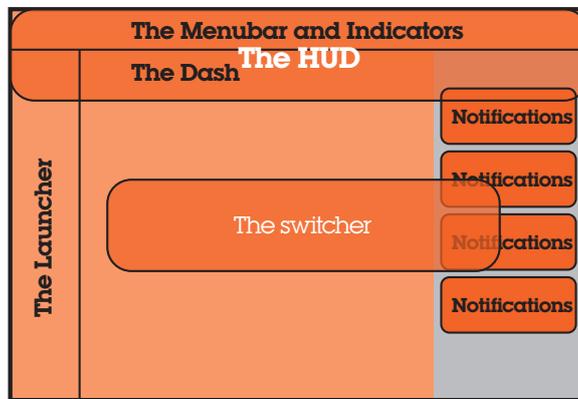


Как им следует перестроить свою работу? Шаттлворт решил, что наилучшим выходом станут релизы Long Term Support, сокращение жизненного цикла промежуточных релизов и «обозначение направления разработки в сторону Rolling Release».

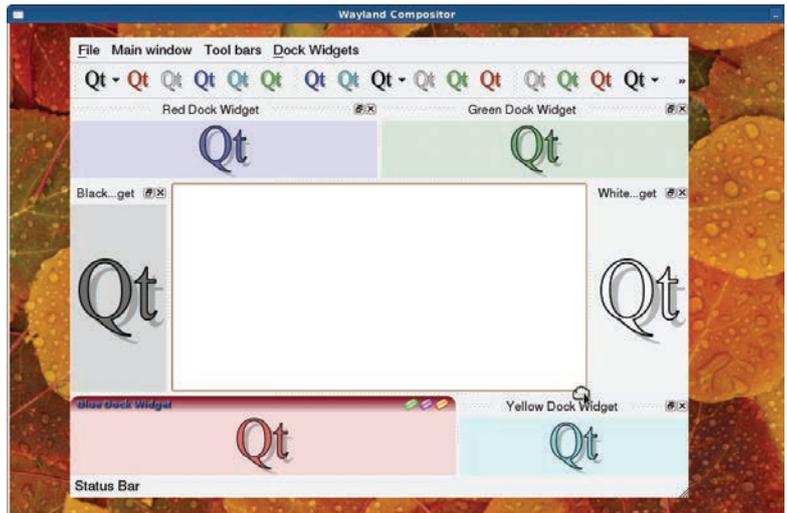
В 2008 Шаттлворт предложил Debian, Gnome, проекту ядра, X и всем остальным проектам приурочить выход их основных релизов к графику разработки Ubuntu в виде «каденции», как он это обозначил. По его словам, «нет ни малейшего сомнения, что чем более сильный “всплеск” мы создадим, скоординировав выпуски и периоды простоя в выходе основных программ из мира свободного ПО, тем сильнее окажется наше влияние на мировой рынок ПО, и тем лучше будет для всех компаний — от MySQL до Alfresco, от Zimbra до IBM, от Red Hat до Ubuntu». На критику Шаттлворт ответил так: «Мы пытаемся подтолкнуть людей к сдвигам не ради нашего удобства. Мы пытаемся ускорить процесс переговоров во всей экосистеме, чтобы все могло разлечься в положенных местах». Многие в сообществе Linux сочли идею координации графиков несерьезной, и от этой концепции отказались.

Космос: последний рубеж

Однако замена Gnome на Unity даже близко не вызвала такого ажиотажа, который вызвало в сообществе анонсирование Mir. Mir — это новый сервер отображения, созданный на замену системы окон X и композитного менеджера Compiz. За этим решением стояло желание Canonical составить конкуренцию операционной системе Android на телефонах и планшетах. X не очень хорошо справлялась с задачей уменьшения масштаба, особенно при наличии некоторых модных эффектов, в то время как Mir, по словам Canonical, в теории предлагает все это в намного меньших, более эффективных и более безопасных пакетах.



› Unity Next — новое поколение Unity, построенное на Qt и предназначенное для работы на Mir — идеально для небольших экранов.



› Два года Ubuntu планировал реализацию Unity на Wayland, системе управления отображением, совместимой со множеством инструментариев, но потом передумал.

Решение перейти на Mir сопровождалось критикой конкурирующего проекта, что вызвало раздражение у многих разработчиков. Инженер-программист Кристиан Хегсберг [Kristian Høgsberg] выступил против заявления Ubuntu, что Wayland не справляется со своей работой. «Технические причины на странице Mir ничего не проясняют... никто не выходит и не говорит всему миру, с чего это Wayland хромает и имеет все проблемы X со вводом».

Дэниел Стоун [Daniel Stone], участник X.Org (одной из базовых технологий, без которых мы бы до сих пор любовались на зеленый текст на черном фоне), пояснил: «Янисколько не беспокоюсь о будущем Wayland. Меня просто злит, что это означает больше работы для нас и больше работы для разработчиков вышедшего ПО, инструментариев и изготовителей оборудования».

Еще один солидный объем кода и бесчисленное количество часов труда разработчиков оказался выброшенным на свалку, но Mir уже начал работать с драйверами дисплея Android и стал, по словам руководства, дыханием будущего. Несмотря на весь плач и скрежет зубовой, Ubuntu неуклонно движется в новом направлении, чего не скажешь о настольных ПК.

В октябре 2012 Ubuntu объявил о том, что больше не станет в том же объеме делиться новостями разработки или релизами разработки с сообществом, как это было раньше. Шаттлворт устал от внутренней борьбы и внешней критики, и заявил: «Хотя мы не будем обсуждать [новые функции] ранее их полной готовности, все же мы рады сотрудничать с членами сообщества, которые доверяют Ubuntu и хотят быть частью разработки».

Что утратил Ubuntu-метр



Смятение среди разработчиков

В недавнем комментарии по поводу Google+ Марк Шаттлворт сказал, что Unity существовала до оболочки Gnome: «Unity, что предельно ясно, была создана первой. Это команда Red Hat топталась на месте, пока не перешла на дизайн, который удивительно похож на Unity».

Адам Уильямсон [Adam Williamson] из Red Hat, который сам себя именуется Fedora QA Community Monkey [Обезьяна-автоответчик], не согласен с этим заявлением, поскольку, по его словам, Gnome 3 впервые обсуждался и планировался на Gnome User Experience Hackfest в 2008. Фактически, Ubuntu несколько месяцев носился с идеей

поддержки Gnome, пока не объявил о собственной идее Unity в 2009.

Еще одной группой разработчиков, выброшенной за борт из-за Unity, оказались люди из Kubuntu. Kubuntu был задуман и реализован для тех, кто предпочитает KDE, а не Gnome. Он получил статус официальной поддержки, а также инфраструктуру и финансирование. На тот момент этого никто не знал, но момент, когда было объявлено о Unity, стал началом конца для Kubuntu. Unity стал первым звонком для всего мира, что Ubuntu начал двигаться в сторону проприетарных мобильных решений.



› Ubuntu Developer Summit, проводимый каждые два года, раньше привлекал сотни разработчиков.

Ubuntu на марше

Окупится ли в долгосрочной перспективе приверженность Canonical к Unity?

Ubuntu продемонстрировал миру то, о чем многие говорили годами: что на самом деле Linux не так уж сложен. Каким-то образом Ubuntu сумел задеть нужные струны и сыграть на революционном духе, совместной собственности и профессиональном статусе для масс. Это было замечательно, это было то, чего многие — например, Mandriva — пытались достичь, но не смогли.

Очень скоро Ubuntu возник на собранных на предприятиях компьютерах. В апреле 2007, через три года после первого появления Ubuntu, Dell объявили о выпуске машин с предустановленным Linux, причем руководитель компании Майкл Делл [Michael Dell] заявил на весь мир, что дома он использует Ubuntu 7.04. Поднявшийся ажиотаж немало поспособствовал росту популярности и повсеместного использования Ubuntu — и Linux в целом.

Linux стал более известен благодаря недолгой безумной популярности нетбуков — небольших, не особо мощных устройств с крошечными клавиатурами, которые не успели толком занять свою нишу на рынке, как их вытеснили оттуда устройства с сенсорными экранами. На многих нетбуках был предустановлен Linux, однако производители просто калечили их — скверно выполненными и кшащими ошибками интерфейсами и дурным подбором программ. На выручку ринулся Ubuntu со своим превосходным Netbook Remix, однако к тому моменту ущерб уже был причинен — многие сочли, что Linux является урезанной дешевой альтернативой Windows.

Это был позор. Как сказал Марк Шаттлворт в 2012 году: «Мы давно знаем, что свободное ПО прекрасно изнутри — оно эффективное, точное, гибкое, его можно изменять. Последние три года мы прилагали усилия, чтобы свободное ПО стало не менее прекрасно и снаружи — чтобы оно стало простым в использовании, приятным для глаз и интересным. А началось это с Ubuntu Netbook Remix, и реализуется в 12.04 LTS». Он был прав, однако у предустановки Ubuntu была парочка фальшстартов.

► **Ubuntu Netbook Remix был весьма достойным дистрибутивом. Увы, он опоздал на вечеринку...**

«Через год примерно 18 миллионов ПК выйдут с Ubuntu на борту.»



► **Canonical надеется войти на переполненный рынок телевизионных развлечений, а также предложить системы для телевидения.**

В 2010 году Dell потихоньку убрала свои компьютеры Ubuntu со своего сайта, заявив: «Мы недавно предприняли попытку упростить наше предложение онлайн, сконцентрировавшись на наших самых популярных партиях и опциях настройки, взяв за основу обратную связь с нашими покупателями, чтобы устранить трудности и создать возможности для совершения покупки самым простым способом... Также мы вносим некоторые изменения в наши страницы Ubuntu, в результате чего они доступны только для покупок по телефону».

Однако не все новости плохи. Превосходный ноутбук XPS 13 от Dell, редкая высококачественная машина с предустановленным Linux, теперь доступна и в США, и в Европе. И Шаттлворт по-прежнему полон оптимизма насчет будущего предустановки Ubuntu. В мае 2012 он заявил на пресс-конференции: «Через год примерно 18 миллионов ПК, или 5 % всей доли рынка, выйдут с предустановленным Ubuntu на борту». На данный момент Canonical заявляет о «20 миллионах пользователей, число которых будет расти» и о 1,3 миллиона сайтов, работающих на серверах Ubuntu, к которым ежемесячно добавляется 22000 новых. Кроме этого, Шаттлворт объявил о планах на Ubuntu TV, Ubuntu-смартфоны и Ubuntu для планшетов. Он прогнозирует, что к 2015 году число пользователей Ubuntu достигнет 200 миллионов, и надеется, что гаджеты — путь, ведущий к достижению этой цифры.

Смена направления

После запуска Ubuntu для телефонов Шаттлворт сказал нам, что ОС была создана с широкой возможностью поддержки устройств, и нам еще предстоит увидеть невероятное количество устройств с сенсорным экраном, работающих на интерфейсе Unity. Мы уже видели одну модель, выход которой планируется на октябрь 2013, однако релиз пока планируется только для Австралии. Подробности ищите на сайте www.ubuntutab.com.au. Что, Ubuntu для телефонов сел на мель или просто его принимают медленнее, чем нам обещали?

Шаттлворт сказал нам при запуске ОС для телефонов, что Canonical поддерживается «очень крупным поставщиком кремния





» «Ubuntu приносит на современные планшеты совершенно новый, потрясающий основы и стильный опыт работы. Он совершенно по-новому подходит к предпрятию», утверждает Canonical.

для производства мобильных телефонов», и что ведутся «переговоры на самом высшем уровне с лидирующими операторами в Северной Америке, Европе и Китае по поиску партнеров для запуска этих устройств».

На той же видеоконференции Шаттлворт озвучил свое видение будущего: универсальный брендинг Ubuntu. Он хочет, чтобы покупатели входили в магазин электроники и мобильных телефонов и сразу узнавали Ubuntu, потому что они уже используют его на своих компьютерах, телефонах, телевизорах и нескольких мобильных устройствах. «Для нас решится множество проблем, если люди будут заходить в магазин и видеть бренд Ubuntu».

Сделано для ТВ

Это было бы солидным достижением, но не преувеличивает ли Шаттлворт, учитывая довольно неутешительный прием, оказанный Ubuntu для телефонов и то, что Ubuntu TV пока что не удается проникнуть на рынок? Ubuntu TV стала первой ОС не для рабочего стола, а для гаджета, вышедшей с интерфейсом Unity, и ее выход сопровождался такими же фанфарами, что и запуск Ubuntu для телефонов. Ubuntu для ТВ должен был совершить революцию в способе потребления контента, предлагая нам свободу смотреть то, что мы хотим, и тогда, когда мы этого хотим. Руко-

водитель Canonical Джейн Силбер [Jane Silber] сказала в момент выхода Ubuntu TV: «OEM и ODM все с большей опаской относятся к подходу типа “запертый сад”, явно применяемому компанией Apple — и которому все чаще пользуется Google, хотя он намного более открыт, чем Apple. Мы наблюдаем колоссальную потребность в нейтральном игроке».

Именно такой подход практикует Canonical с Ubuntu для телефонов: что есть совместная монополия Apple и Google на смартфоны и прочие небольшие мобильные устройства, и что OEM всего мира прямо-таки рыдают о системе, которая наконец-то вызовет их из-под диктата большой двойки — о «нейтральном игроке», как его определила Силбер.

Мы, конечно, надеемся на дальнейший успех телефонов и планшетов, но не поздно ли для Ubuntu влиять на рынок? Если поздно, и если специальные функции для телефона и планшета, которые, по слухам, должны быть в предстоящем релизе Ubuntu (загляните на <https://wiki.ubuntu.com/Touch/Install>, чтобы найти там копию Ubuntu Touch Developer Preview — но сначала прочтите предупреждение) будут в основном проигнорированы, то пора задуматься, стоило ли таких усилий стремление Canonical к конвергенции. Спустя год после запуска Ubuntu TV число вовлеченных изготовителей равно нулю.

«OEM с большей опаской относятся к подходу “запертый сад”»

Что утратил Ubuntu-метр



Куда идти дальше?

Одна идея, которая может установить гармонию в отношениях Ubuntu с сообществом, заключается в том, чтобы дать Ubuntu свободу. Поскольку Canonical поддерживает инфраструктуру официальных ответвлений Ubuntu, возможно, Canonical могли бы поддержать фонд Ubuntu и отделить коммерческое направление от свободного. Конечно же, это уже делалось раньше, и вполне успешно, и имеется

несколько моделей, из которых можно выбрать. Модели Red Hat/Fedora и SUSE/openSUSE — вот идея того, как спонсорская компания может брать код и даже направлять разработку. И хотя эта ситуация не для всех разработчиков идеальна, она позволяет направить деятельность пользователей на отчеты об ошибках. Эта модель действительно позволяет возвращать программистов и создавать прекрасную операционную систему для сообщества. Однако для Canonical она может представлять собой проблему. Видите ли, им же приходится иметь дело с этими противными собственниками, с «акционерами», как их именует Шаттлворт.

Возможно, Canonical могли бы прибегнуть к гибридной открытой системе. Canonical могли бы предоставить фонду инфраструктуру, свободу и возможность

использования какой-то разновидности имени Ubuntu. Это вряд ли будет дорого стоить, и у сообщества будет собственный дистрибутив. Затем Canonical могли бы оплатить своим штатным разработчикам реализацию коммерческих устремлений, нужных Ubuntu, не обременяясь ответственностью перед общественностью и оппозицией. А потом сообщество может сколько угодно орать на разработчиков версии сообщества, если кнопки вдруг куда-то переместятся или поиск на рабочем столе вдруг станет осуществляться совместно с Amazon.

Canonical больше не пришлось бы беспокоиться по поводу открытого кода, а все вопросы по открытому коду направлялись бы в «OpenUbuntu Foundation». И тогда они с полным правом могли бы заявить: «Смотрите — мы тоже вносим свой вклад, посмотрите на OpenUbuntu Foundation». И это также могло бы примирить и настольные дистрибутивы на базе Ubuntu, и Canonical смогут прекратить поддерживать пакеты X, да и все остальное, от чего они с удовольствием бы избавились.



» Семейство устройств Ubuntu в представлении Шаттлворта — «полная конвергенция; одно семейство интерфейсов, одна кодовая база».

Чем Ubuntu велик

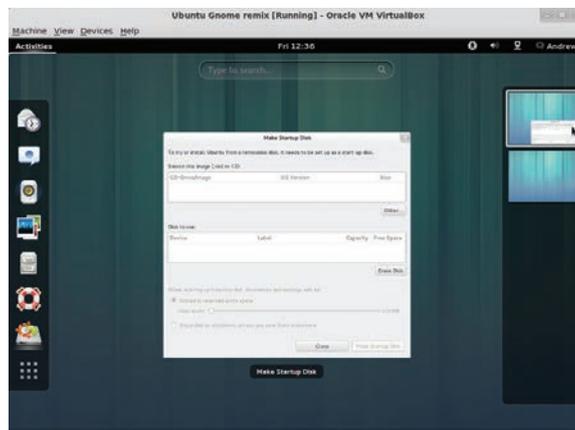
Бен Эверард снимает шляпу перед Шаттлвортом за то, что тот не боится рисковать.

Десять лет назад мир Linux был непохож на сегодняшний. В те времена сама идея использования Linux кем-то кроме компьютерных гениев казалась смехотворной. Марк Шаттлворт все изменил. Он выдвинул идею о том, что Linux может быть прост в применении — и сделал его таким. Выше мы сказали, что Canonical предоставил ядру Linux меньше заплат, чем другие компании. Причина в том, что в Canonical осознали то, чего не поняли другие: пользователям рабочего стола нет дела до ядра — им нужен хорошо интегрированный, простой в использовании рабочий стол. Именно на это Canonical и тратит свое время; и знаете что? Им удалось привлечь к Linux больше пользователей, чем Red Hat, SUSE или прочим солидным разработчикам ядра. Вкладом Canonical в экосистему Linux стал не новый код, а новый образ мысли. Образ мысли, который говорит: «Эй, Linux годен для всех».

И что интересно, Linux именно в таком образе мысли тогда и нуждался. Над ядром уже трудились сотни разработчиков, но лишь немногие думали над тем, чтобы сделать его дружелюбным.

Второе, что сделал Canonical для Linux — предоставил ему инфраструктуру. Возьмем, например, Linux Mint. Это один из самых популярных, если не самый популярный на сегодняшний день дистрибутив Linux. Его можно легко считать последователем Ubuntu в качестве Linux для простого смертного. Но разработчики Mint сумели сконцентрироваться на создании максимального комфорта для пользователя, придя на готовенькое. Они просто использовали репозитории Ubuntu, а затем безо всяких усилий получили серверы, распространяющие их обновления по всему миру безвозмездно.

Ubuntu также единственный из оставшихся основных коммерческих дистрибутивов, которыми реально могут пользоваться обычные люди. Конечно, Red Hat делает все эти потрясающие вещи; вероятно, это просто чудесный дистрибутив; но вы никогда



➤ Если вам не нравится Unity, всегда есть Lubuntu, Xubuntu, Kubuntu или Ubuntu GNOME Remix, как видно здесь.

этого не узнаете, если не готовы потратить на него несколько тысяч фунтов. То же относится к SUSE. Да, у обоих есть версии сообщества, но лучшие функции они сохраняют только для корпоративных клиентов. Причина в том, что все они (Canonical, Red Hat и SUSE) являются коммерческими, а не благотворительными организациями. Будь мир совершенным, Canonical предоставлял бы работу большому числу разработчиков ядра и продвигал все их изменения прародителям, а Red Hat Enterprise Linux можно было бы скачивать бесплатно (да, мы знаем, что CentOS свободна, но это как раз не благодаря, а вопреки усилиям Red Hat). Суть в том, что в каждой компании Linux приходится идти на компромиссы.

Прошло уже почти десять лет с момента запуска Ubuntu, и рабочий стол Linux медленно, но верно улучшается. В этом нет ничего плохого, но иногда нужна хорошенькая встряска, сделать которую с согласия сообщества нельзя. Как говорил Генри Форд: «Спроси я людей, что им нужно, они ответили бы: лошадей побыстрее». Чтобы переступить через определенный порог, нужно, чтобы кто-то его перешагнул и рассказал, как это делается. И, оглядываясь назад, мы либо хвалим этих людей как провидцев, либо осуждаем, называя их глупцами; однако в момент действий, пока не осядет пыль, их всегда будут считать смутьянами.

Unity как раз и является такой встряской. Он заставил нас задуматься о том, что такое хороший рабочий стол и о том, используем ли мы свои компьютеры наилучшим способом. И даже если в итоге Unity ждет провал, все же мы как сообщество станем мудрее в плане создания хороших интерфейсов. Огромное количество опций рабочего стола в Linux означает возможность их попробовать, не лишая пользователей других возможностей. В конце концов, это не Canonical прекратила работу над GNOME 2, а сам GNOME. Если вам не нравится Unity, то KDE, Xfce, LXDE и (как в самом последнем релизе) GNOME Shell — все они имеют официальные версии Ubuntu.

Unity грациозно подвел нас к Mir, еще одному спорному графическому проекту Canonical. Мы не собираемся оправдывать Canonical: с их стороны было неверным ходом распространять

«Unity заставил нас думать, что такое хороший рабочий стол.»

➤ Благодаря Сиппапон, Mint принял у Ubuntu эстетику удобства в использовании.



дезинформацию о Wayland, когда они анонсировали Mir. Мы надеемся, что это было искреннее заблуждение, а не сознательная попытка распространить страх, неуверенность и сомнения в альтернативной системе, но даже если это случилось по ошибке, в Canonical должны были знать, что информация будет распространена весьма широко, и они должны были тщательно проверить точность фактов.

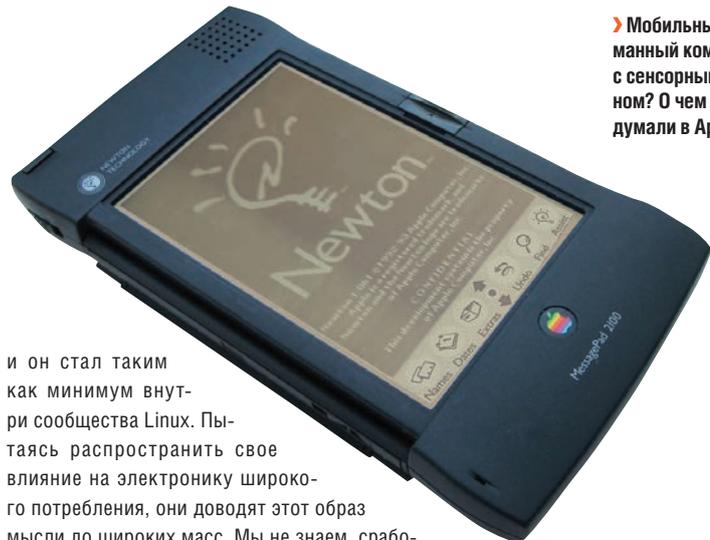
Однако наличие непроверенной информации в публичном анонсе вовсе не означает, что вся идея этого анонса неверная. Нас не посвятили в причины, почему Canonical решили, что им это надо. Но раз уж они так решили, они подошли к этому с точки зрения наименьшего количества повторяющейся работы для приложений. Это реализация Qt и связок GTK, и предоставление поддержки преемственности для X. Будем надеяться, это означает, что вся дополнительная работа упадет на Canonical.

Вне рабочего стола

В Apple выпустили планшет. Возможно, вы слышали о нем — он назывался Newton. Честно говоря, мира он не перевернул и постепенно канул в забвение, и его производство прекратилось в 1998 году. Естественно, в Apple поняли, что они не могут выпускать мобильные устройства, и прекратили свои попытки? Случись такое, мы бы сейчас жили в совершенно ином мире. Возможно, мы бы все использовали Blackberries; возможно, в Nokia в конечном итоге решились бы на выпуск телефона на базе Linux; а может быть, мы бы все использовали обычные мобильники с батареями, держащими заряд по две недели, и играли бы в *Snake 2*. Нам этого никогда не узнать, потому что преуспевающие компании ведут себя не так. И успешные люди тоже. Если мы хотим, чтобы Linux добился успеха, нам нельзя требовать, чтобы разработчики Linux шли у нас на поводу.

Ubuntu TV пока не добился успеха, но это не означает, что сама идея была плоха, и уж точно не означает, что Canonical следует перестать пробовать что-то новое. Применение Ubuntu в электронных устройствах широкого потребления стало бы колоссальным благом для всего движения открытого кода. Позвольте объяснить, почему. Android построен на Linux. Мы все знаем это, однако то, что находится поверх ядра Linux — это смесь проприетарных программных центров (Google Play и т. д.) и коммерческих «приложений». В основном это коммерческая система, разве что основанная на ядре с открытым кодом. Но, что еще более важно, она не угрожает стабильным игрокам на любом рынке, занимающимся чем-либо помимо мобильных устройств. Она не заставляет людей использовать Linux на чем-либо, кроме своих телефонов. А теперь представьте на минуту, что появились телефоны Ubuntu. Представьте, что они стали третьим игроком после iPhone и телефонов на Android. Представьте, что вы приходите в магазин мобильных телефонов и видите там широкий ассортимент телефонов Ubuntu, а продавцы демонстрируют вам, как ими пользоваться. Возможно, в этом потенциальном будущем для вас не будет никакой разницы. В конце концов, тот факт, что вы читаете *Linux Format*, означает, что вы, вероятно, уже один из нас. Однако представьте себе, что это будет значить для людей, менее разбирающихся в технике. Для людей, которые — если они вообще слышали про Ubuntu — полагают, что это исключительно для умников. И вдруг его начнут рассматривать как дружелюбную к пользователю — а то и клевою — систему. Если это произойдет, то на настольный Ubuntu тоже распространится такое восприятие, и их начнут считать потенциальными компьютерами для всех.

Десять лет назад Canonical смогли создать образ мысли, выражающийся фразой «Эй, Linux может быть доступен каждому»,



► Мобильный карманный компьютер с сенсорным экраном? О чем вообще думали в Apple?

и он стал таким как минимум внутри сообщества Linux. Пытаясь распространить свое влияние на электронику широкого потребления, они доводят этот образ мысли до широких масс. Мы не знаем, сработает ли это, но аплодируем им уже за попытку. Если это сработает, цепная реакция для других дистрибутивов будет крайне обширна. Не в последнюю очередь потому, что настольный Linux будет рассматриваться как основная платформа, и его станут воспринимать всерьез производители оборудования.

Компьютеры изменяются, и изменяются быстро. Раньше они стояли на наших столах, а теперь они находятся в облаке и в наших карманах.

Возможно, завтра они будут у нас в очках и часах. Время и прилив не ждут никого, и то же можно сказать об изменениях технологий. Чтобы избежать риска, следует оставить будущее новаторам и позволить догнать себя в бесконечной игре в догонялки.

С точки зрения настольного Linux, избегать риска означает навсегда остаться продуктом для технарей и чудачков-любителей. Применение новшеств означает шанс определять будущее, но это неизбежно связано с риском потери ваших постоянных пользователей.

Уникальное преимущество настольного Linux в том, что мы можем делать и то, и другое. Инновациями займется Ubuntu, в то время как Mint, OpenSUSE, и все остальные будут избегать риска. В этом преимущество экосистемы открытого кода, которую мы создавали годами. Однако мы можем пользоваться этим преимуществом только в том случае, если люди готовы идти на риск. Марк Шаттлворт вступил в большую игру. Если он выиграет — выиграет все сообщество Linux. **LXF**

«Ubuntu начнут рассматривать как дружелюбную систему.»

Скажите свое слово

Вы один из тех тысяч, которые сбежали с Ubuntu на Linux Mint? Вы пытались работать с Ubuntu в мрачные дни сомнительной поддержки оборудования и шли с ним через огонь и воду? Вы экономите каждую копейку и ждете, когда закончится действие вашего контракта, чтобы вложить деньги в телефон или планшет Ubuntu? Или, может быть, вы все эти годы используете Debian или Slackware и не понимаете всей этой

болтовни вокруг Ubuntu? Кем бы вы ни были, мы хотим услышать ваше мнение по поводу последних разработок этого новаторского дистрибутива. Оставьте свой комментарий на форумах Linux Format (www.linuxformat.com/forums) или отправьте нам сообщение на lf.letters@futurenet.co.uk и поделитесь с нами своими мыслями. Лучшие ответы появятся на страницах, посвященных письмам, в будущем номере.





Наш вклад в мобильное будущее

Mozilla надеется, что свободная мобильная платформа Firefox OS откроет новые горизонты.

LXF беседует с ведущим инженером Джонасом Сикингом.



В мире Linux каждый год начинается с волны пророчеств стать «годом чего-то» — как правило, настольного. И как правило, все они оказываются преувеличенными. Однако 2013-й может стать исключением. С тремя новыми мобильными ОС — Firefox, Ubuntu и Sailfish — готовыми к запуску, нас, возможно, наконец-то ждет год Linux-телефона (в смысле, кроме Android). На момент написания этой статьи первые образцы на базе одной из трех только начали поступать в продажу: устройства от Geeksphone на Firefox OS. Мы обратились к Джонасу Сикингу [Jonas Sicking], работающему над мобильным предложением Mozilla, дабы узнать, чего ждать...

LXF: Как лично вы пришли к открытому ПО?

ДжС: Это давняя история! Когда в 1998-м Netscape стал открытым, я подумал: «Ну, круто». Я начал бродить по сайту, присматриваясь к проекту Mozilla, и наткнулся на исходный код. И вдруг: «Э, да я это понимаю!» Я тогда был веб-разработчиком, и мне хотелось применить одну технологию. Был некий код для ее поддержки, но фактически нерабочий. Думаю: «А что если довести его до ума?» Идея увлекала меня все



больше и больше, и вскоре я попал на практику в Netscape. Окончив школу, я поступил в Mozilla, где и работаю до сих пор.

LXF: И как там атмосфера?

ДжС: Не устаю наслаждаться, хоть я и давно уж здесь. Здесь очень инженерный подход. Правда, сотрудничать с производителями мобильных

телефонов нам в диковинку. Выпуская код в этой сфере, приходится гораздо плотнее взаимодействовать с партнерами, да еще всякие внешние ограничения, которые идут не от наших инженеров, — все это очень непривычно.

Обычно в Mozilla, если ты хочешь что-либо разработать, ты обращаешься к сообществу, и если им твоя идея нравится, берешься за дело.



О КОНКУРЕНЦИИ

«Возможно, выживет только модель, которая окажется лучше.»

Инициатива идет снизу вверх, и для нас очень важно делать все правильно. Исключена мотивация «надо сделать, потому что это принесет деньги» или нечто подобное. Все ориентировано на инженерный результат, и очень открыто. Вы можете пойти и поговорить с кем угодно.

LXF: Идея с созданием телефона родилась внутри Mozilla, или это был запрос производителей?

ДжС: Это было давно ожидаемый шаг. Браузер медленно эволюционировал от средства для просмотра документов до системы, под которую пишут приложения, и логично предположить, что сейчас он уже вполне может стать основой для операционной системы. В сообществе заговорили о том, что понадобится, чтобы превратить Firefox в ОС, и в какой-то момент обсуждения вылились в «Давайте попробуем!» До написания самого когда дело дошло года полтора назад.

LXF: Входит ли в планы создание настольного компьютера для OS?

ДжС: А как же. Проект Firefox OS делится на две части: программная платформа, позволяющая работать с приложениями и собственно делающая систему ощутимой; и компоненты, необходимые для создания ОС, такие как взаимодействие с телефоном и настройка Wi-Fi.

Приложения задуманы абсолютно кросс-платформенными. На самом деле, мы начали с разработки настольной версии, но когда заметили, насколько выше сейчас интерес ко всему мобиль-

ному — очевидно, за этим будущее — мы решили сконцентрироваться на мобильных приложениях. Так и появилась Firefox OS — этот путь был ей предначертан. Система выполнения, используемая нами в приложениях, также подходит для Firefox Android и настольной Firefox, так что можно написать одно в расчете на все три. А в перспективе, помимо Firefox OS, оно будет запускаться на Chrome OS и Tizen, а также исходно на Android.

Телефоны Firefox OS

Первые телефоны на Firefox OS только поступили в продажу. Компания Geeksphone выпустила две модели, Keon и Peak.

В сердце обоих — процессоры Qualcomm Snapdragon: в Keon — более медленный одноядерный чип, в Peak — более быстрый двоядерный. В обоих 512 МБ ОЗУ и 4 ГБ на диске, но у Keon 3,5-дюймовый экран, а у Peak более широкий

4,3-дюймовый. Подробную информацию вы найдете на www.geeksphone.com. Если мы раздразили ваш аппетит, вам не повезло: все они были мгновенно распроданы.

Однако для особо жаждущих есть другой способ опробовать новую ОС. Устройство на Firefox от Sony еще не готово, зато они выпустили ПЗУ с Firefox OS для Xperia E.

LXF: Только что были анонсированы три новых открытых мобильных платформы — по-вашему, это будет взаимное дополнение или конкуренция?

ДжС: По-моему, это эксперимент, — посмотрим, что из этого сработает, что понравится пользователям; не исключено, что они дополняют друг друга, или, возможно, выживет только модель, которая окажется лучше.

Одна из наших целей с Firefox OS — то, что приложения должны быть кросс-платформенными. Чтобы не надо было писать отдельно вариант для iOS, для Android, для Firefox OS... Должно быть просто одно приложение, которое запускается везде. Многие из тех, кто также строит мобильные платформы, интересуются и web-приложениями; мы стараемся с ними сотрудничать, чтобы делать все по единому стандарту.

LXF: Значит, вы работаете с командами из других организаций?

ДжС: Да. Мы ведем переговоры с Ubuntu — пока неясно, что из этого получится. Сотрудничаем, например, с Samsung, поскольку у них есть платформа Tizen (см. рубрику *Что за штука?* на стр. 58), чтобы наши приложения были доступны и там; и привлекаем Google, чтобы обеспечить работу также в ChromeOS.

LXF: Мы слышали, что вы нацелены на бюджетные модели телефонов. Вы считаете эту часть рынка более перспективной, или это долгосрочный стратегический расчет?

ДжС: Я не эксперт в области рынка, но по-моему, именно сейчас это перспективно. Разумеется, речь не идет о долгосрочной стратегии — ведь web-технологии также применимы и к моделям высокого класса. И дело не в том, что мы пока не можем лучше: просто сейчас открыта именно эта ниша рынка, именно здесь мы нашли своих партнеров. Запускать новую платформу всегда рискованно, а на бюджетных моделях можно сэкономить.

О КРОСС-ПЛАТФОРМЕННОСТИ

«Должно быть просто одно приложение, которое запускается везде.»

LXF: А как производители оборудования приняли идею о создании мобильного телефона Firefox?

ДжС: Да прекрасно! Найти партнеров Mozilla не составило труда — как в плане оборудования, так и его наполнения. По-моему, все сходится на том, что будущее — за web-приложениями, вопрос лишь в том, кому удастся первым создать более достойную платформу. И мне кажется, у Firefox OS есть шанс, поскольку у Mozilla неплохая репутация в сетевом мире. Люди верят, что у нас все получится.

LXF: Бытует мнение, что JavaScript — и web-приложения в целом — работают медленнее, чем системные. Столкнулись ли вы с этим на бюджетных моделях?

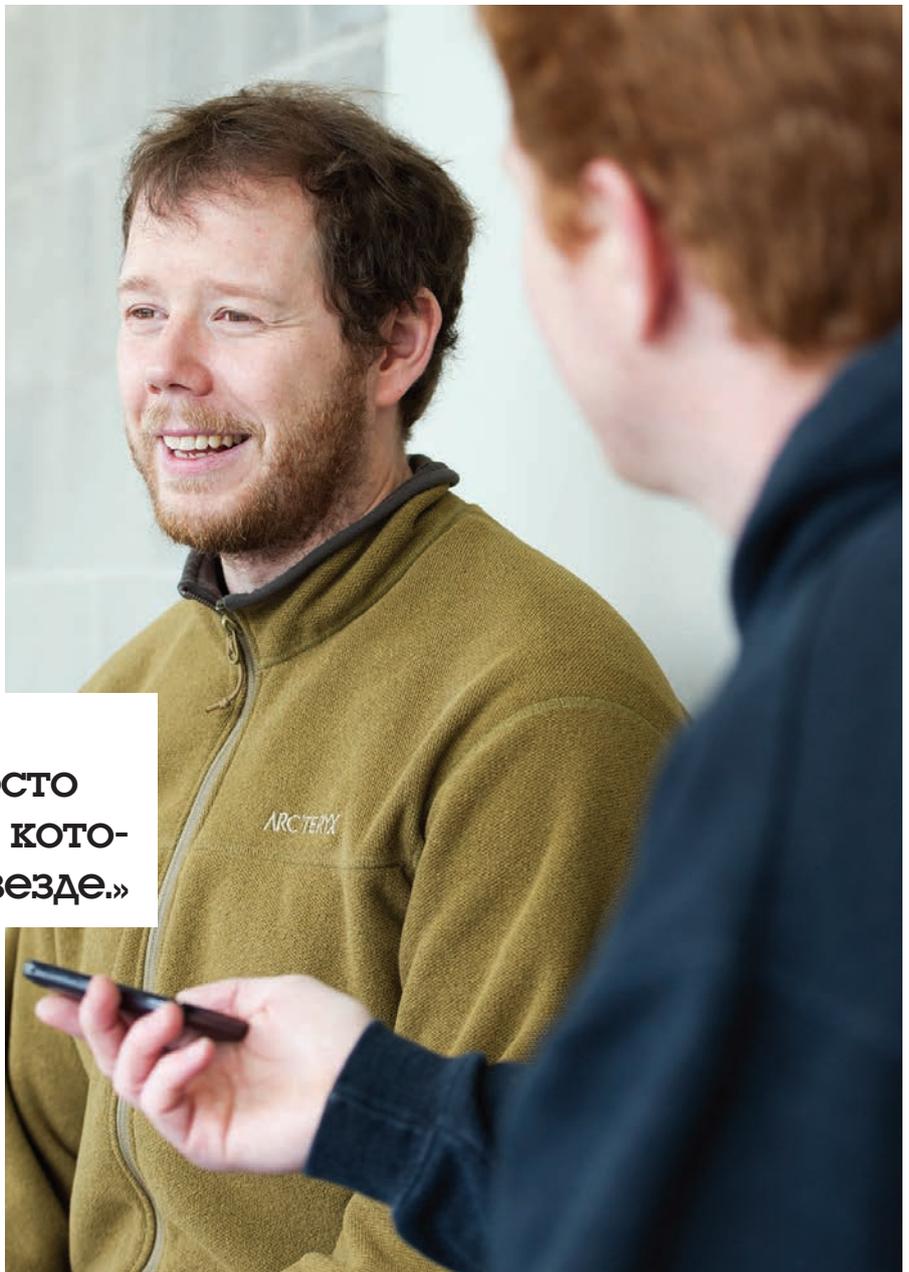
ДжС: В отношении скорости приложений, логика C++ не действует. Если взять крутую игру, решающим фактором чаще всего будет игровой движок — он отвечает за рендеринг и прочее. Если речь об офисных приложениях, например, сетевых — опять же, важна не логика, а такие вещи, как IO, сеть и так далее. JavaScript становится очень конкурентоспособным. Естественно, кое-что дается с трудом, но по большей части, как ни странно, все получается — так что это не сдерживающий фактор.

LXF: Чем вы особенно гордитесь?

ДжС: Тем, что это не просто открытый проект — во многом мы занимаемся созданием открытой

платформы. Где каждый может, к примеру, создать собственный магазин. Можно даже не размещать там свое приложение, а просто раскрыть при помощи сайта. Хорошо, что это стало, наконец, доступно.

Радует также и наше внимание к безопасности. Мы действительно постарались, чтобы ощущение защищенности в Интернете перешло и на приложения, и можно было без опаски обращаться в любой магазин и устанавливать что угодно. Если же приложению требуются какие-то приватные данные, мы обращаемся к самому пользователю. Количество вопросов невелико, и сформулированы они внятно. Мы не спрашиваем такие вещи, как «Хотите ли вы предоставить доступ к TCP-сокету» или даже «доступ к сети» — людям неясно, что это значит. А вот вопрос «Хотите ли вы предоставить приложению доступ к вашим картинкам?» поймут все. **LXF**





Trisquel изнутри

Соратник Free Software Foundation по борьбе за свободу ПО.

Есть четкая граница между идеологией и прагматизмом. Проект Trisquel GNU/Linux, возможно, единственный дистрибутив, умудрившийся устоять на этой тонкой черте. Дистрибутив Trisquel — самый практичный дистрибутив, который придерживается жестких правил Free Software Foundation (FSF) без ущерба удобству в использовании для рядового пользователя настольной системы.

Проект Trisquel был запущен студентами Университета Виго [Vigo] в Испании, которые изначально хотели создать образовательную операционную систему с поддержкой галисийского языка. Но скоро этот проект перерос свою исходную цель, и Trisquel стал полностью

свободной многоязычной операционной системой. Сейчас у этого дистрибутива есть сильное международное сообщество, состоящее из добровольцев, возглавляемое Рубеном Родригесом [Rubén Rodríguez]. Trisquel Linux был официально представлен в апреле 2005 года самим Ричардом Столлменом [Richard Stallman].

Выступая на LibrePlanet 2013 в Массачусетсе, Родригес сказал, что, по его мнению, распространение свободного ПО налагает большую

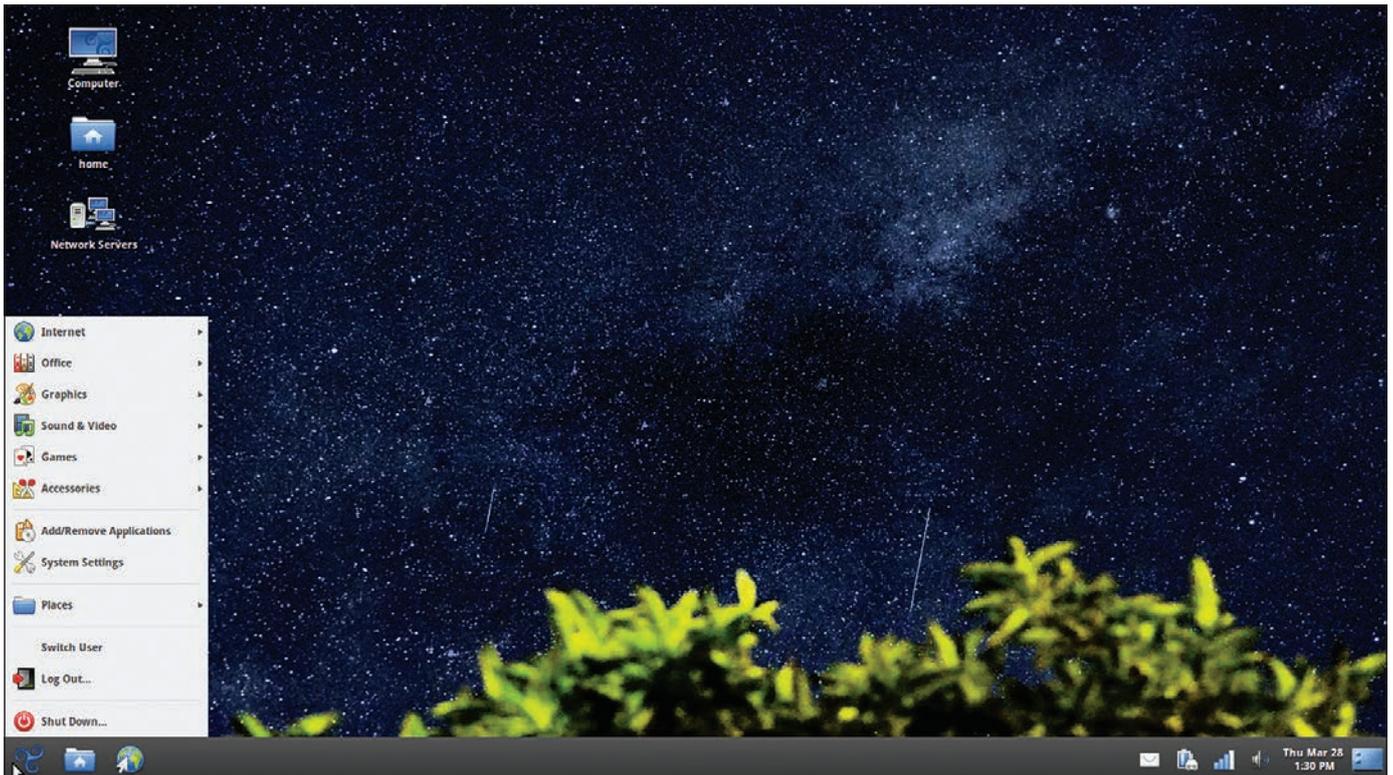
ответственность: «Ваши пользователи зависят от вас [считая], что вы все делаете правильно и [что вы будете] уважать их свободу».

Именно поэтому проект уделяет такое пристальное внимание концепции дистрибутива. «Мы не используем слово “дистрибутив” как техническое определение программного проекта — как пакет или tarball. Для нас слово “дистрибутив” означает ответственность за наших пользователей. По этой причине мы включаем только свободное ПО в наши образы и репозитории», объяснил Родригес.

«Дистрибутив» означает ответственность за пользователей.»

Свободный Ubuntu?

Для непривычного взгляда может показаться, что Trisquel — это просто Ubuntu



минус несвободные пакеты, которые не согласуются с жесткими принципами свободного ПО GNU, однако Родригес с жаром уверяет, что не все так просто.

«Чрезмерно упрощая, можно подумать, что сделать свободный дистрибутив — это выкинуть кое-что из Ubuntu, и все, — говорит он. — Однако отцедить оттуда несвободные программы — задача не из легких. Иногда их надо извлечь, ничего при этом не сломав».

В качестве примера Родригес приводит работу ядра Linux-libre. Цель проекта Linux-libre в том, чтобы взять модифицированную версию ядра и удалить все программы, которые не включают исходный код или выпускают его под проприетарными лицензиями.

Если у драйвера есть и свободная, и несвободная версии, большинство дистрибутивов-прототипов, таких, как Ubuntu и Fedora/Red Hat, предпочитают несвободную. Но когда разработчики пытаются ее удалить, они должны на ее место поставить свободную версию, не повредив никаких других компонентов дистрибутива.

Цель команды Trisquel не только в том, чтобы сделать дистрибутив «свободнее» — они также надеются улучшить и другие

области. Джонатан Надо [Jonathan Nadeau] из FSF сотрудничает с разработчиками Trisquel, чтобы сделать дистрибутив более доступным для слабовидящих пользователей.

Фактически, главное в последнем релизе, Trisquel 6.0, это его улучшенные функции доступности. По умолчанию рабочий стол загружается со включенной доступностью, и программа для озвучивания текста с экрана *Orca* знакомит пользователей с интерфейсом.

Еще одна цель проекта Trisquel — внешне украсить рабочий стол и сделать его проще в навигации. Вообще-то Trisquel с виду не очень отличается от Windows, что, по словам Родригеса, должно облегчить переход с одной ОС на другую. «Нет ничего плохого в том, чтобы скопировать нечто успешное», добавил он.

Libre внутри

Trisquel Linux имеется в четырех редакциях. Основной релиз — это 700-МБ Live CD (устанавливаемый), который использует рабочий стол Gnome и идет с английским и испанским переводом.

» FSF раздает копии Trisquel на самых разных мероприятиях, в том числе на запуске Windows 8 в Нью-Йорк Сити.

»

Использование Trisquel 6.0

Trisquel 6.0 — самый последний релиз дистрибутива; он основан на Ubuntu 12.04. Подобно дистрибутиву-родителю, Trisquel 6.0 — релиз с долгосрочной поддержкой (Long Term Support, LTS); он предлагает отладки ошибок и обновления системы безопасности до 2017 года.

Лучшее в релизе — его привычный интерфейс рабочего стола. Хотя дистрибутив работает на Gnome 3.4, разработчики — по причинам, изложенным в статье — используют резервный интерфейс, чтобы обеспечить классическое общение с рабочим столом.

Основная редакция дистрибутива снабжена LibreOffice 3.5 и версией 19 web-браузера Abrowser.

Вместо проприетарного Flash-плагина в нем имеется свободный медиа-плеер *Gnash*, и он умеет воспроизводить HTML5-видео и отображать файлы PDF без плагина.

Для установки добавочных программ вы можете либо воспользоваться *Synaptic*, либо собственным менеджером пакетов дистрибутива. В репозиториях дистрибутива есть множество свободных программ, таких, как *VLC Player*, позволяющий воспроизводить самые разные виды мультимедиа-файлов.

Для установки дистрибутив использует модифицированную версию программы установки Ubuntu, которая является одной из самых простых программ установки для пользователей рабочего стола.

И опять-таки, если вы хотите установить несвободные приложения, например, *Skype*, вы можете сделать это с помощью *Synaptic*.

Помните, что дистрибутив не включает никаких проприетарных драйверов, что может вызвать определенные проблемы с оборудованием, особенно с беспроводными картами. С другой стороны, состояние свободных драйверов за последние годы значительно улучшилось, и на наших машинах Trisquel работал без всяких проблем.

Даже если вы не фанат свободы, Trisquel 6.0 — очень привлекательное производное Ubuntu, которое работает куда лучше большинства дистрибутивов на базе Ubuntu, о которых мы рассказывали.

Затем имеется 500-МБ версия Mini, с рабочим столом LXDE, предназначенная для нетбуков и более старого оборудования. Есть также 1,2-ГБ версия i18n, которая идет с переводом на 50 языков, и 25-МБ версия NetInstall, которая во время установки скачивает пакеты из Интернета. Все редакции доступны для 32- и 64-битной архитектуры.

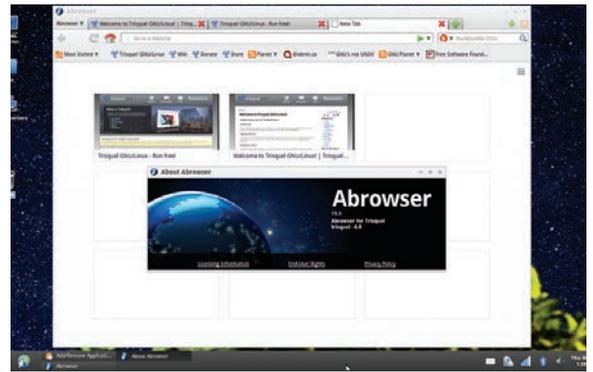
Хотя Trisquel предлагает обычный набор лучших свободных программ, таких, как *LibreOffice* и *Gnome*, он также замещает некоторые компоненты, чтобы соответствовать своей цели. Две самых интересных программы, которых вы не найдете в популярных дистрибутивах — это web-браузер Trisquel, *Abrowser*, и ядро *Linux-libre*.

Abrowser — это не *Firefox* для Trisquel без бренда Mozilla. Команда Trisquel модифицировала браузер так, что тот теперь рекомендует только свободные дополнения и плагины. Проект также ведет список расширений, тем и переводов Libre на <https://trisquel.info/en/browser>.

Trisquel обязан поддерживать совместимость с остальной частью системы (на базе Ubuntu), и вместо того, чтобы брать простое ядро Linux, он применяет модифицированную версию Ubuntu. Однако прежде чем разработчики могут за нее взяться, им приходится устранять все несвободные двоичные куски. Это достигается скриптом *deblob* из проекта *Linux-libre* для соответствующей версии ядра. Затем разработчики запускают это ядро со скриптом *Linux-libre*, чтобы добавить самые последние функции.

Как указывает на своем сайте команда Trisquel, «в результате этой процедуры мы не только устраняем несвободные куски, но также удостоверяемся, что ни один фрагмент ядра не потребует для своей работы несвободной программы».

«Ни один фрагмент ядра не потребует несвободной программы.»



► Безбрендовый web-браузер Trisquel берет только свободные дополнения из собственного репозитория Trisquel.

Правила GNU

Сборка релиза Trisquel отличается от остальных производных дистрибутивов, потому что этот релиз должен соответствовать требованиям GNU для свободных дистрибутивов (см. врезку «Свободный дистрибутив — что это?» внизу). Работа над новым релизом Trisquel начинается сразу после выхода релиза Ubuntu. Весь процесс чистки Ubuntu обычно занимает до шести месяцев, и примерно в это время объявляется о выходе релиза Trisquel.

Первая задача — создание репозитория пакетов разработки. Разработчики создают локальное зеркало последнего релиза Ubuntu, и оно обновляется каждую ночь с помощью инструмента *Reprepro* и некоторых скриптов *Cron*. Затем пакеты в этом зеркале чистят, сверяясь с черным списком несвободных пакетов. Очищенный репозиторий уже считается настоящим родительским репозиторием, над которым работают программисты Trisquel.

Свободный дистрибутив — что это?

Trisquel является свободным дистрибутивом. Что это на самом деле означает?

В соответствии с законом о копирайте, создатель работы — единственное лицо, обладающее такими правами, как право изменять или распространять свою работу. Если говорить о лицензиях на программные продукты, свободное ПО предоставляет это право всем своим пользователям. Обычно лицензия свободного ПО обеспечивает четыре права:

- 1 Свобода использовать ПО.
- 2 Свобода изучать ПО и вносить изменения для удовлетворения собственных потребностей. Реализация этой свободы требует доступа к исходному коду ПО.
- 3 Свобода распространять копии ПО.
- 4 Свобода распространять копии вашей измененной версии ПО.

Самая популярная лицензия Free Software — это лицензия GNU GPL License, которая и гарантирует эти четыре права. Сравните: хотя вы платите за проприетарное ПО и приобретаете официальную лицензионную копию, вы все же не получаете свобод 2, 3 и 4 — таким образом, ваша плата взимается только за использование ПО.

Учитывая все это, естественно будет признать, что дистрибутив, который включает только свободное

ПО, должен быть свободным дистрибутивом. Однако это неверно, потому что дистрибутив — не просто набор программ. Программы, конечно же, основной компонент, но есть также шрифты, документация и другие элементы, такие, как библиотеки, определяющие взаимодействие ПО с оборудованием, и т.д.

Чтобы дистрибутив был признан свободным, каждый из этих функциональных компонентов должен быть доступен в виде исходника, то есть исходный код каждого из этих компонентов должен быть доступен пользователям. Дополнительно, каждый из этих компонентов должен также быть доступен под одной из множества лицензий свободного ПО.

В дополнение к упомянутым (обычно именуемым «функциональными компонентами данных» или «информацией для практического использования»), дистрибутив также имеет и другие компоненты. Эти элементы, такие как оформление или иные художественные компоненты, способствующие созданию эстетической стороны восприятия, а не функциональности, не обязательно должны быть доступны под лицензией свободного ПО. Поскольку художественные компоненты разрешены к свободному распространению как в коммерческих, так и в некоммерческих целях, их можно включить в свободный дистрибутив.

Одно из довольно суровых требований к свободному дистрибутиву — он ни при каких обстоятельствах не должен отсылать своих пользователей к несвободным компонентам. Среди этих «несвободных элементов» могут быть драйверы для оборудования, или любая иная программа, например, расширение для web-браузера.

Trisquel довольно жестко следует этим опубликованным правилам FSF. Перед появлением релиза дистрибутив проходит стадию тщательной чистки, во время которой все его компоненты проверяются на наличие проприетарных фрагментов. Все, что не соответствует правилам, удаляется или, если это необходимо, заменяется на свободные эквиваленты.

По этой причине в таких дистрибутивах, как Debian, *Firefox* выходит под названием *Iceweasel*: разработчики хотят предоставить пользователям браузер, который не включает никаких несвободных компонентов. Жесткая интерпретация правил FSF командой Trisquel привела к тому, что разработчики пошли на шаг дальше: вместо *Iceweasel*, Trisquel идет с *Abrowser*, который размещается на их собственном сервере и, хотя тоже основан на *Firefox*, не поощряет пользователей устанавливать несвободные дополнения.

В дополнение к вышележащим пакетам, этот репозиторий также содержит замещения отвергнутым пакетам, такие как ядро Linux-libre, и пакеты, которые требуют ребрендинга, как веб-браузер, а также набор специальных художественных пакетов для Trisquel, файлы настройки и метапакеты.

Разработчики начинают трудиться над этим репозиторием, который затем тестируется обновлением набора тестовых машин. Если все в порядке, то пакеты размещаются в официальных репозиториях с помощью *rsync*.

Сложности с пакетами

Самая сложная и интересная часть процесса разработки — это создание настраиваемых пакетов Trisquel. Это делается с помощью набора скриптов, именуемых *helpers* [помощники]. По словам Родригеса, они похожи на инструмент *Builder*, созданный командой дистрибутива *gNewSense* — это еще один дистрибутивом *Free Software*, одобренный *FSF*.

Trisquel использует скрипты-помощники для изменения и компиляции некоторых пакетов из репозитория *Ubuntu*, потому что они либо содержат несвободные компоненты, либо разработчики хотят их настроить.

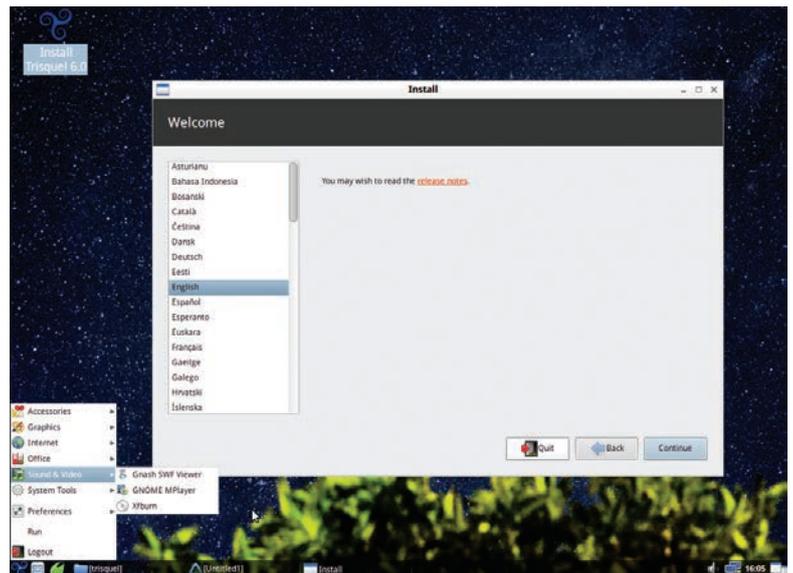
По своей сути, помощники — небольшие скрипты *Bash*, которые содержат сниппеты и регулярные выражения потокового редактора *Sed*. Разработчики применяют их, чтобы слегка модифицировать пакеты, которые во всем остальном вполне хороши и пригодны к использованию.

«Скажем, вы хотите изменить родительский пакет, улучшив его, — объясняет Родригес. — Например, устранить то, чего там не должно быть, тем самым сделав его свободнее, или убрать какую-то фразу, которая предлагает использовать несвободные программы, или изменить что-нибудь интеллектуальной собственностью, или подобную же чепуху».

В таком случае помощник берет исходный код — обычно из *Ubuntu*, но создатели пакетов Trisquel могут указать и другой источник, например, PPA или репозиторий *Debian* — чтобы сделать версию, более соответствующую Trisquel. Получив код, скрипт-помощник подготовит его к комментариям, следующим после него, чтобы изменить пакет.

«При необходимости он построит всю систему компиляции, — объясняет Родригес. — Поэтому он создает *chroot jails*, в которых происходит компиляция, вносит внутрь код, создает пакеты, подписывает их и помещает в репозитории разработки Trisquel».

«Команда всегда ищет добровольцев, желающих помочь.»



Чтобы создать окончательный ISO-образ, Trisquel использует инструмент *debootstrap* внутри скрипта, который также устанавливает метапакет для создания разных редакций дистрибутива. Затем он сжимает их в образ *squashfs* и делает его загружаемым с помощью *Isolinux*. Проект использует настраиваемый скрипт под названием *makeditro* для создания набора DVD исходного кода.

Руки свободны?

Подобно любым проектам *Free Software*, разрабатываемым сообществом, команда Trisquel всегда ищет добровольцев, желающих помочь улучшить дистрибутив и устранить из него ошибки. По словам Родригеса, одной из причин, по которым дистрибутив работает на некоторых системах с проблемами, в том, что проекту не хватает достойной тестовой инфраструктуры. «Нам требуется всеобъемлющий список того, что нужно протестировать в каждом образе Trisquel, который мы выпускаем», сказал он.

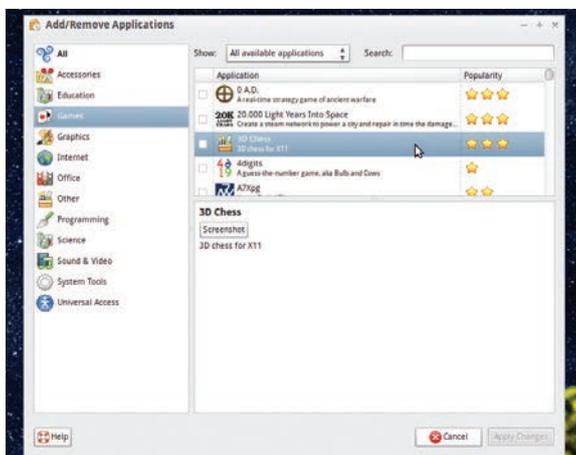
Родригес заявил, что одна из самых кошмарных ошибок программ — резервная панель *Gnome 3*, портированная из *Gnome 2*, но разработчики Trisquel работают с проектом *Gnome*, чтобы поддержать жизнь в режиме *fallback*. Другие проекты отказались от *Gnome 3* из-за его радикально нового пользовательского интерфейса, но для Trisquel основная причина не использовать *Gnome Shell* в том, что она неудобна для пользователей с ограниченными зрением, и навигацию в ней нельзя осуществлять полностью с клавиатуры.

Также *Gnome Shell* требует наличия 3D-ускорения, а текущее состояние свободных драйверов для ускорения графики оставляет желать лучшего.

Если вы не разработчик, но все же хотите принять участие в проекте Trisquel, вы можете сделать это или с помощью программы участия в разработке дистрибутива, или сделав взнос (<http://trisquel.info>). «Мы запустили систему членства после того, как начали делать образ, который поставляется с членской картой *Free Software Foundation*, и мы этим очень гордимся», сказал Родригес.

Trisquel — это дистрибутив, который предлагает все удобства *Ubuntu*, но разработчики поставили во главу угла вашу свободу и конфиденциальность. Trisquel GNU/Linux являет собой доказательство — если таковые вообще нужны — того, что вполне возможно продуктивно работать в совершенно свободной операционной системе. **LXF**

► **Mini-редакция Trisquel использует рабочий стол LXDE и идеальна для машин с ограниченными ресурсами.**



► С помощью менеджера пакетов Trisquel можно установить только свободные программы — для установки несвободных обращайтесь к *Synaptic*.



Властелин былого

Отправьтесь по тропе памяти вместе с **Джоно Бэконом** и узнайте, как построить в *Gramps* свое генеалогическое древо.



Как правило, люди знают своих ближайших родственников — хоть им порой и не хочется их знать! — но как насчет прабабушек и прадедушек? Или пра-пра? Где жили ваши предки? Чем занимались? И как получилось, что вы оказались именно там, где оказались? Генеалогия — это наука об исследовании, отслеживании и создании генеалогических деревьев, и она должна помочь вам найти ответы на некоторые из этих вопросов.

Генеалогия — типично исследовательский проект. Весь фокус в том,

чтобы найти надежные источники, способные помочь вам в создании исторически точного генеалогического древа. В столь интенсивной исследовательской работе — ознакомлении с многочисленными публикациями, интервью, книгами и прочими носителями — может оказаться трудным не запутаться во всех записях. Вот здесь-то и пригодится *Gramps* — всеоб-

емлющая генеалогическая программа с открытым кодом, которая работает на Linux, Mac и Windows.

Gramps был написан специалистами в генеалогии, и можно смело сказать, что это — один из лучших инструментов, чтобы справиться с этой работой. Вы можете найти его в пакетах большинства дистрибутивов, или скачать с <http://gramps-project.org/download/>.

Прежде чем начать, уделим пару минут обсуждению того, как отслеживать генеалогическое древо и какие практические приемы мы будем

**«Нельзя помещать
в древо то, что ничем
не подтверждено.»**

Хранение файлов

Если вы сохраняете в вашем генеалогическом древе фотографии людей, важно помнить, что *Gramps* сохраняет не сам носитель, а ссылку на его местонахождение на вашем жестком диске. Вам следует выбрать директорию для хранения ваших изображений, видео, аудио и документов. Затем вы можете настроить эту директорию как стандартное расположение, выбрав Edit > Preferences и вкладку General. Установите в поле Base Path For Relative Media [Путь к медиа] указание на эту директорию и щелкните по Close, чтобы сохранить изменения.

щелкните по вкладке Gallery. Теперь щелкните по кнопке Add, чтобы добавить новое изображение.

Сначала вам предложат выбрать носитель, который вы собираетесь отобразить, и затем вы увидите Media Reference Editor. В верхней части окна щелкните по вкладке Source Citations и укажите источник изображения — помните, источник должен быть у всего!

Добавьте события

Далее надо добавить день рождения Джона Смита; это основное событие в его жизни, и мы должны зафиксировать его. Откройте Джон Смит в виде Person и щелкните по вкладке Events [События] (она должна быть выделена). Теперь щелкните по значку Add. Gramps предложит вам Birth [Рождение] в качестве первого события, которое вы захотите добавить, поэтому введите день рождения Джона в виде MM/DD/YYYY [Месяц/Дата/Год] (например, 08/30/1979) или Day Month Year [День Месяц Год] (30 Авг 1979). Если вы используете Day Month Year, месяц должен указываться только первыми тремя буквами или полностью. Также можно щелкнуть по кнопке Calendar [Календарь] справа, чтобы ввести даты.

Теперь нужно добавить место рождения Джона, но мы пока не фиксировали этого в нашем проекте. Щелкните по значку Add после поля Place. Введите места в виде City/Town [Город], Township [Область], County [Округ], State [Штат], Country [Страна] или разные сочетания указанных компонентов. Если вы укажете широту и долготу, это позволит вам впоследствии расположить эти места на карте.

Навязчивый голосок у вас в голове уже должен твердить: «Укажите источник этой информации!». Щелкните по вкладке Source, откройте Clipboard, перетащите ваш источник (интервью), выберите уровень Confidence и сохраните источник и новое место с помощью кнопок OK.

Добавьте родителей

Далее мы должны добавить родителей Джона. Выберите значок на панели инструментов Add A New Set Of Parents, щелкните по кнопке Add рядом с Mother and Father [Мать и Отец] и введите тот же тип информации, что и раньше, то есть

для этого использовать. Создание генеалогического древа предполагает три основных шага:

1 Исследование Найдите информацию о вашей семье — например, где жили ваши предки или когда они вступали в брак.

2 Запись Занесите любую новую информацию.

3 Публикация Публикация отчетов и записей, которыми вы готовы поделиться с другими.

Вероятно, больше всего времени вы потратите на два первых шага — исследование и постоянное обновление вашего древа, а затем вы периодически будете делать отчеты. К счастью, *Gramps* облегчает любой из этих трех шагов.

Золотое правило генеалогии гласит, что вы всегда должны быть в состоянии записать, как и где вы нашли информацию. Создать генеалогическое древо может кто угодно, основываясь на догадках, но если вы хотите сделать эту работу должным образом, ни в коем случае нельзя помещать в древо то, что ничем не подтверждено.

Приступаем

При первой загрузке *Gramps* появится окно. В нашей статье мы отследим генеалогическое древо вымышленной семьи Смит — она намного родоветее моей!

Щелкните по кнопке New [Создать] в диалоговом окне Family Trees [Фамильные древа] и добавьте в качестве имени «Семья Смит». Теперь выберите Load Family Tree [Загрузить фамильное древо], и перед вами предстанет главное окно *Gramps*. Слева вы увидите список информационных категорий, таких как People [Люди], Relationships [Отношения], Families [Семьи] и т.д. Если щелкнуть по любой из них, основной вид и панели инструментов изменятся, чтобы отобразить тип информации. Именно так вы конструируете разные ветви своей семьи.

В большинстве генеалогических древ вы начинаете с самого себя, но здесь мы начнем с Джона Смита, который и осуществляет исследование.

Создайте источник

Поскольку мы начинаем с Джона, нашим первым источником будет интервью Джона с самим собой.

Ваш контент должен, как правило, исходить из независимых достоверных источников, но если вы просто хотите добавить информацию о себе и ближайших членах своей семьи, интервью с самим собой в качестве источника вполне приемлемо. Щелкните в левой боковой панели по значку Source [Источник] и выберите Add [Добавить] в панели инструментов, чтобы открыть Source Editor [Редактор источников].

Чтобы создать источник, добавьте «Интервью с Джоном Смитом» в поле Title [Заголовок] и «Джон Смит» в поле Author [Автор]. Теперь щелкните OK для завершения.

Затем добавим в источник некоторую информацию. Дважды щелкните по источнику, чтобы открыть Интервью с Джоном Смитом. Откроется Source Editor. Щелкните по вкладке Notes и выберите Add, и перед вами предстанет текстовый редактор, где можно добавлять полное содержание интервью с самим собой.

Создайте личность

Суть генеалогического древа — люди, так что добавим нашу первую личность. Вы угадали правильно: это Джон Смит.

Поскольку вся информация, которую мы добавляем, идет из источника «Интервью с Джоном Смитом», вначале нам нужно переместить наш источник в легкодоступное место, чтобы мы могли использовать данные в разных отображениях. На панели инструментов щелкните по значку Clipboard [буфер обмена] и перетащите «Интервью с Джоном Смитом» на него из вида Source. Теперь ваш источник хранится в Clipboard, и вы можете закрыть окно.

Теперь выберите вид Person [Личность] и щелкните по кнопке Add на панели инструментов, чтобы ввести информацию о вашей первой личности. Добавьте следующую информацию о Джоне:

» Имя: Джон

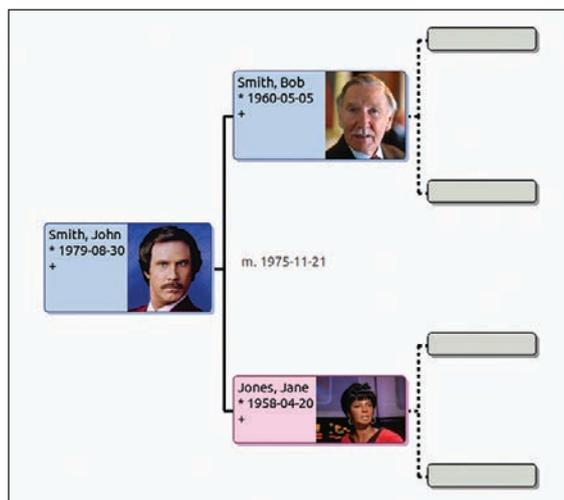
» Фамилия: Смит

» Пол: Мужской

Затем вам нужно сослаться на ваш источник. Щелкните по вкладке Sources, затем откройте Clipboard из панели инструментов и перетащите источник, который вы сохранили там, на область под ID | Title | Auth | Page в Person Editor [Редактор личностей].

Откроется Source Reference Editor [Редактор ссылок на источники]. Нижняя часть — уже введенный источник, но верхняя часть — новая: она содержит данные о том, как была обнаружена информация и насколько вы убеждены в ее достоверности. Для данного интервью вы можете настроить достоверность на Very High [очень высокая], а также указать дату.

Чтобы оживить наше древо, обвешаем его фотографиями всех членов семьи. В виде Person щелкните по записи Джон Смит, чтобы загрузить редактор Person, затем



» В редакторе *Gramps* вы можете видеть разные поколения.

имя, день рождения, место рождения и т. д., — не забывая об источниках. Добавив обоих родителей, щелкните по выпадающему меню Type, чтобы выбрать их отношения, в данном случае — Married [В браке]. Теперь мы добавили обоих родителей и указали, что они состоят в браке, но не указали дату, место и прочие подробности их брака. Это событие, имеющее отношение к семье, и мы можем указать его в Family Editor.

Щелкните по виду Families и дважды — по записи. Теперь щелкните по вкладке Events и затем Add. Выберите Marriage [Брак] из выпадающего меню типа события и затем укажите дату события (и, конечно, источник).

А теперь давайте предположим, что, к несчастью, родители Джона развелись. Щелкните по таблице Events в Family Editor и выберите Add, но на этот раз выберите Divorce [Развод]. Добавьте дату развода — и источник!

Добавьте братьев и сестер

Оказывается, Джон не единственный ребенок в семье Смит — у него есть два брата, Мартин и Саймон. Давайте добавим Мартина в качестве родного брата (он одинок) и Саймона в качестве родного брата, у которого есть девушка.

Чтобы добавить обоих братьев, щелкните по семье в виде Family и выберите вкладку Children [Дети]. Теперь щелкните по кнопке Add и введите их имена и информацию о рождении. Когда вы нажмете ОК, чтобы добавить данные о родном брате или сестре, у вас спросят об их отношении к матери и отцу — это позволит уточнить, является ли ребенок родным, приемным или пасынком/падчерицей, и т. д.

Мартин-то одинок, но Саймон состоит в современных отношениях с девушкой, и его девушку

зовут Лора Джейкобс. Давайте добавим Лору в наше генеалогическое древо. Добавление партнеров в Gramps несколько необычно, хотя и несложно. Сначала щелкните по виду Relationship [Отношения] и выберите Саймон Смит. Теперь щелкните по Edit > Add Partner [Редактировать > Добавить партнера]. Как ни странно, диалоговое окно укажет Саймона как отца (даже при том, что никаких детей у него нет), но в любом случае вы можете добавить Лору в поле Mother [Мать] в диалоговом окне, а также указать, когда и где она родилась — и источник!

Добавление Лоры в генеалогическое древо придает некую уникальность — новую связь между семейством Смит и другой семьей. Добавив Лору, можете добавлять других членов ее семьи, используя уже описанные нами методы.

Вы можете добавить родителей Лоры, выбрав Лору в виде Relationship и затем щелкнув по кнопке на панели инструментов Add A New Set Of Parents [Добавить новый набор родителей]. Аналогично можно добавлять других братьев и сестер, детей и других родственников, а также всю информацию, события и источники, относящиеся к ним.

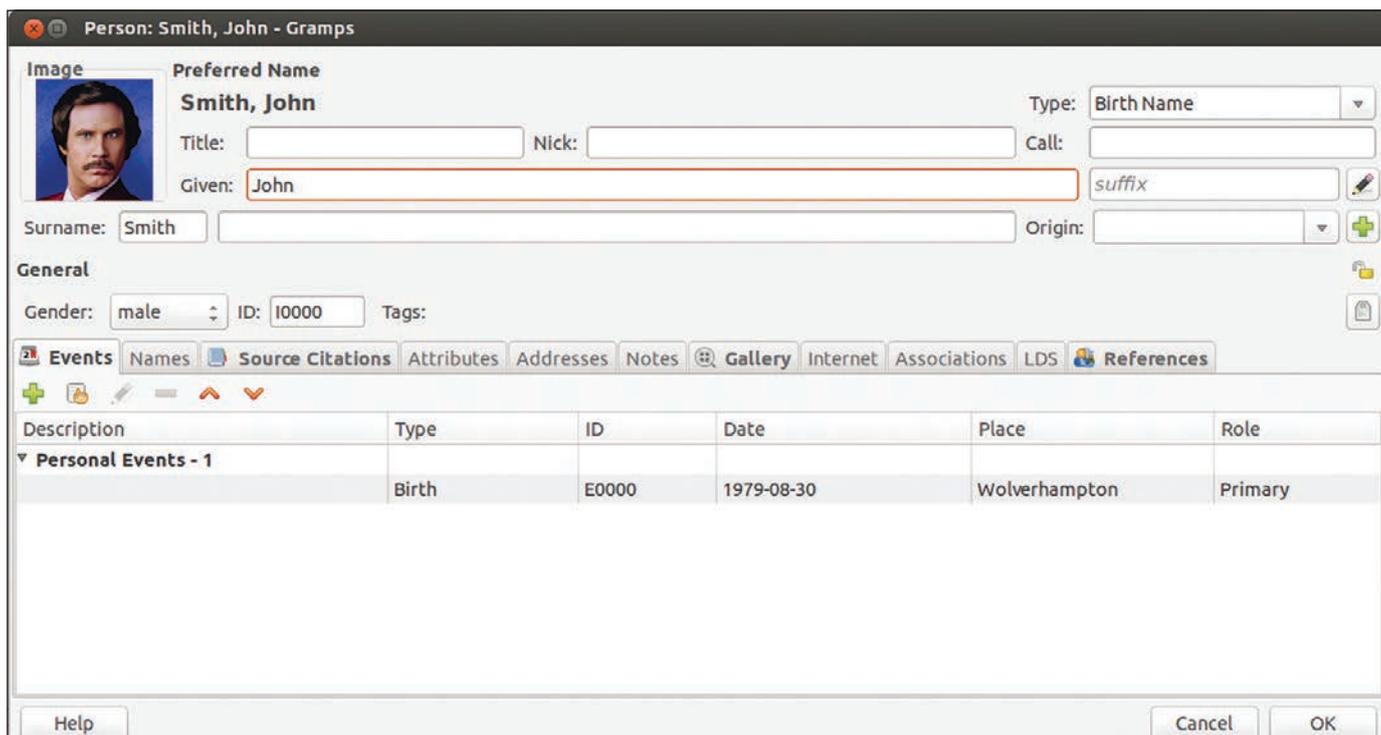
Растим древо

Мы рассказали об основных функциях Gramps для добавления разных членов вашей семьи и отслеживания информации об их жизни. Теперь



главное — пойти и отследить как можно больше членов вашей семьи.

Эта задача потребует от вас почти детективного расследования в стиле CSI. Где бы вы ни обнаружили новую семейную связь, думайте о всяких местах, где нашлась бы более подробная информация о ней. Если обнаруженное вами самое старшее поколение — пра-пра-прабабушки и пра-пра-прадедушки, попробуйте выяснить, кем были их родители. Здесь-то и кроется основная прелесть генеалогии: попытки обнаружить достоверные источники позволят вам все глубже



➤ Gramps позволяет хранить всю информацию о членах семьи — например, где и когда они родились — в одном месте.

погрузиться в историю своей семьи. К источникам, достойным внимания, относятся:

- » данные налоговых органов;
- » записи в домовых книгах;
- » медицинские записи;
- » полицейские и судебные записи (кто знает, вдруг среди ваших родственников имелись и негодяи);
- » газетные статьи и публикации;
- » интервью с друзьями семьи, коллегами и партнерами.

Хотя все больше записей можно найти онлайн, всегда стоит сходить в библиотеку и архив — значимая информация об истории вашей семьи часто обнаруживается именно там.

Создание отчетов

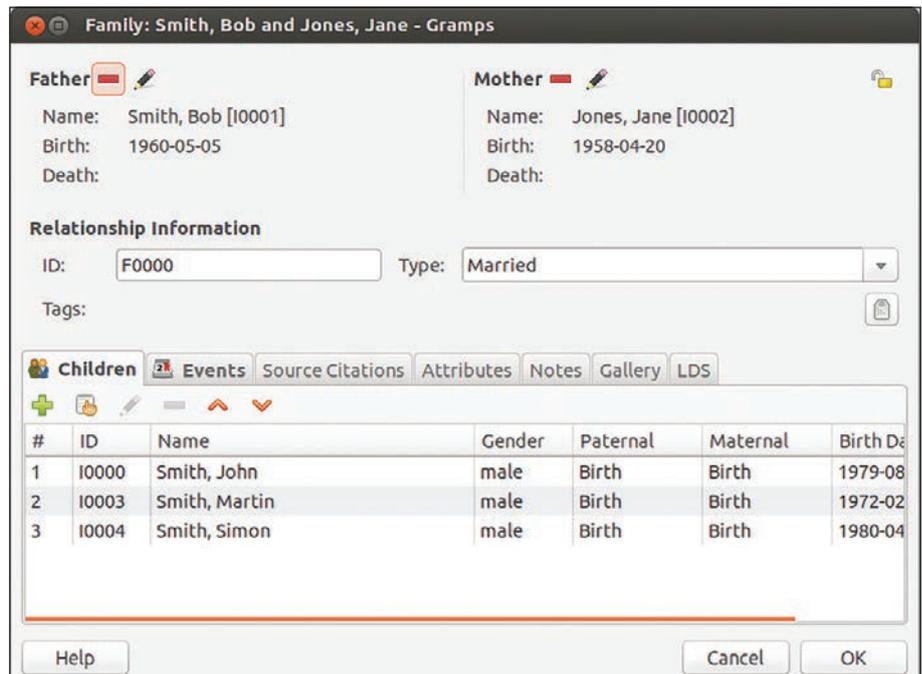
Когда у вас получится довольно развесистое генеалогическое древо, вам захочется представлять информацию разными способами, в зависимости от того, кому вы ее показываете.

На текущий момент все ваше исследование размещается внутри редактора, но *Gramps* также включает всесторонние функции для трансформации этих данных в разные виды отчетов — отдельных или являющихся частью книги. Это очень полезный способ поделиться результатами своей работы с семьей и друзьями.

В *Gramps* доступен ряд разных форматов отчетов; мы охотно рассказали бы обо всех, но место есть только на основные.

Начнем с текстовых форматов: отчеты, которые предоставляют текстовый итог различных аспектов вашего генеалогического древа. *Gramps* предоставляет богатый выбор, включая:

- » **Ahnentafel Report** [Отчет о предках] — признанная генеалогическая система нумерации (Ahnentafel — нем. предок)
- » **Birthday and Anniversary Report** [Отчет о днях рождения и годовщинах] — та же информация, что и в календаре, но в текстовом формате
- » **Complete Individual Report** [Полный индивидуальный отчет]
- » **Descendant Report** [Отчет о потомках]
- » **Detailed Ancestral Report** [Подробный отчет о предках]
- » **Detailed Descendant Report** [Подробный отчет о потомках]
- » **End of Line Report** [Отчет о конце линии] — список последних известных предков
- » **Family Group Report** [Отчет о семейных группах]
- » **Kinship Report** [Отчет о родстве]
- » **Tag Report** [Отчет о тэгах] — первоначальные объекты (личности, семьи, заметки), которые соответствуют выбранному тэгу
- » **Number of Ancestors** [Количество предков]



» Отслеживать знаменательные события, даты и места очень легко с помощью Family Editor.

- » **Place Report** [Отчет о месте]
- » **Records Report** [Отчет о записях] — интересные записи (в основном относящиеся к возрасту), такие как старейший из живых родственников, самая молодая мать, и т.д.
- » **Summary of the Database** [Сводка базы данных]

Большинство из этих отчетов создается щелчком по Reports > Text Reports и выбором нужного

«Всегда стоит сходить в библиотеку и архив.»

вам типа. Вы также можете настроить особые аспекты формата: например, для Complete Individual Report можно выбрать, какую информацию включать в сводку. *Gramps* также предусматривает ряд графических отчетов, в том числе:

- » **Ancestor Tree** [Древо предков]
- » **Calendar** [Календарь]
- » **Descendant Tree** [Древо потомков]
- » **Family Descendant Tree** [Древо семейных предков]
- » **Fan Chart** [Веерная диаграмма]
- » **Statistics Chart** [Статистический график]
- » **Timeline Chart** [Временной график]

И опять, вы можете выбрать эти отчеты из Reports > Graphical Reports и настроить разные

аспекты результата. Например, выбрав Timeline Chart, вы можете указать порядок отображения графика. Многие из этих отчетов генерируются в виде файлов *LibreOffice*.

Один из конечных форматов отчета, о котором стоит упомянуть — книжный отчет [book report]. Он создает полную книгу, сводящую воедино всю нужную информацию из вашего генеалогического древа. Чтобы создать книгу, щелкните по Reports > Books > Book Report. В диалоговом окне введите название книги и затем выберите разные разделы из окна Available Items и щелкните по Add, чтобы добавить их в раздел Current Book. Затем вы можете изменить порядок разделов с помощью кнопок со стрелками и нажать OK, чтобы создать книгу.

Друг семьи

Gramps — феноменально мощная программа, разработанная и созданная специалистами в области генеалогии для специалистов в области генеалогии. Хотя сначала *Gramps* может вызвать некое замешательство — при столь огромном количестве функций, опций и кнопок — она дает все необходимые возможности для создания полного отчета, где будет указана подробнейшая информация о вашей семье, которую вы смогли нарыть, и вся она будет располагаться в удобном для вас месте.

Являетесь ли вы любителем, который хочет получить общее представление о своем генеалогическом древе, или профессионалом в области генеалогии, который проводит исследование по чьему-либо заказу, *Gramps* предоставит вам все необходимые инструменты.

Удачи в исследовании вашего генеалогического древа, и не забудьте рассказать нам в Linux Format о своих результатах. LXF

Настройка отчетов

Если вы хотите настроить внешний вид и работу множества разных отчетов, которые может создавать *Gramps*, щелкните по кнопке Style Editor в диалоговом окне отчетов.

Войдя в редактор, щелкните по кнопке Add, чтобы добавить новый профиль, затем выберите пункты на левой панели и используйте вкладки для настройки стилей.

Lords of the Board

Александр Хрюкин твердо решил обзавестись мини-компьютером на базе процессора ARM. И вот что из этого вышло...

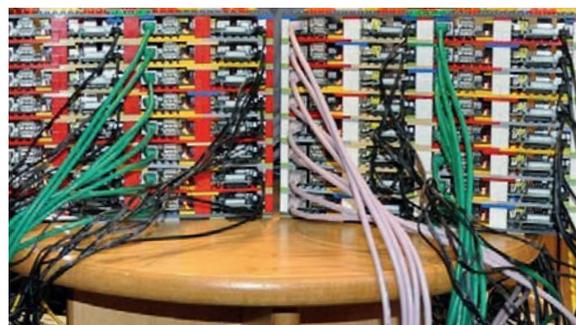
6 мая 2011 года Дэвид Брейбен [David Braben] представил первый концепт Raspberry Pi размером чуть больше обычного USB-флэш-накопителя. Мне тот день запомнился очень хорошо, потому что на многих радиоловительских, программистских и других околокомпьютерных ресурсах началась просто какая-то истерика в соответствующих темах обсуждений — к слову сказать, достаточно быстро растягивающихся на 40–70 страниц. То и дело люди спрашивали, где можно купить это устройство или хотя бы записаться в очередь на предзаказ.

И вот 10 января 2012 года Raspberry Pi Foundation вроде бы как условно выпустила партию в 10 000 штук, которая разлетелась с такой скоростью, что мне показалось, будто бы ничего и не выпускалось. Вот спецификации Raspberry Pi:

ЦПУ	700-МГц ARM11 ARM1176JZF-S core
Графический процессор	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264 high-profile кодирование/декодирование
ОЗУ	(SDRAM) 256 MiB или 512 MiB
Порты USB 2.0	1 (поставлен BCM2835) или 2 (через интегрированный USB-хаб)
Видеовыводы	Composite video Composite RCA, HDMI (не одновременно)
Выходы аудио	TRS-коннектор гнездо 3,5 мм, HDMI
Входы аудио	нет, но можно добавить микрофон или звуковую карту через USB
Встроенное хранилище	Secure Digital/SD / MMC / SDBвод/Вывод слот для карты

Примерно в то же время я попытался заказать ее через различные ресурсы, которые указаны на <http://www.raspberrypi.org/> в разделе дистрибуции [Buy a Pi]. Однако везде они либо закончились, либо предлагались по предзаказу к сентябрю-октябрю (напоминаю, что на дворе был конец января), и кроме того, больше двух девайсов в одни руки было попросту не купить.

Я решил плюнуть на всю эту затею и подождать, пока вокруг Raspberry Pi не уляжется шум и товарищи из Поднебесной



➤ Тот самый кластер из Raspberry Pi. Одним все, другим ничего...

не наштмпуют своих собственных одноплатных компьютеров по цене 100 баксов за мешок.

Потянулись долгие дни ожидания, и в сентябре 2012 я увидел чудную новость на opennet.ru, в рамках которой демонстрировались счастливые лица энтузиастов из Саутгемптонского университета на фоне кластера из 64 одноплатных компьютеров Raspberry Pi — <http://www.opennet.ru/opennews/art.shtml?num=34823>. Тут-то и стало окончательно очевидно, что дистрибуцией компьютеров занимаются каким-то особенно хитрым образом: некоторые могут приобрести устройства аж в количестве 64 штук, тогда как остальные торчат в очереди за дефицитным товаром с мая по октябрь.

Сил ждать больше не было, а заявления про две штуки в одни руки и ожидание на полгода на фоне кластеров выглядело просто издевательством.

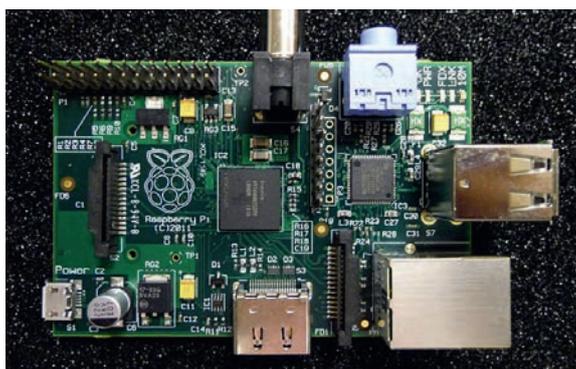
Сам для себя я решил, что буду смотреть в сторону других устройств, которые могут быть чуть дороже, но главное — чтобы они имелись в доступе.

Первым делом на глаза мне попала такая забавная штука, как МК802 от компании Miniand (видеообзор этого устройства был опубликован на LXF172).

И, что характерно, эта штука оказалась значительно мощнее, чем RPi:

ЦПУ	Allwinner A10 1.0 ГГц Cortex-A8 + 500-МГц GPU
Графический процессор	2D / 3D / OpenGL ES2.0 (AMD Z430) / OpenVG1.1 (AMD Z160) @ 27M Tri/s
ОС	Android 4.0
ОЗУ	512 МБ / 1 ГБ DDR3 (фактически доступная память несколько меньше из-за резервирования)
Внутренняя память	Nand Flash 4 ГБ (1 ГБ используется системой)
Сеть	Wi-Fi 802.11b/g, WAPI (Ralink8188)
Хранилище	Внешнее через USB и microSD

➤ Появление Raspberry Pi ознаменовало бум одноплатных мини-компьютеров.





➤ MK802 — плата от компании Miniland.

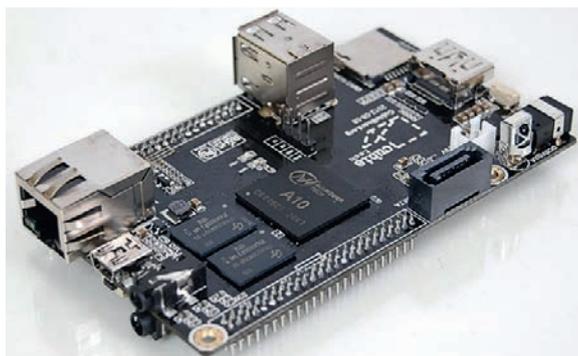
Все выглядело достаточно симпатично, и я заказал себе именно такое устройство. По стоимости вышло около \$50, что вполне сопоставимо с ценой в \$35 на RPi, ведь там доставка оплачивалась еще отдельно. Век высоких технологий — он такой! Цена почтового отправления сравнима с ценой высокотехнологичной электроники.

Огорчало в MK802 только отсутствие Ethernet-порта и тот факт, что вместо беспроводного адаптера использовалось ралинковская поделка. В остальном нареканий не было. Мне удалось запустить на нем Debian Testing, Cyanogenmod, Fedora16.

Та же компания, что выпустила MK802, у себя на страничке демонстрирует еще несколько интересных устройств на базе процессора allwinner (cortex a-8).

Это Hackberry A10 Dev Board:

ЦПУ	1,2-ГГц Allwinner A10 ARM Cortex A8
Графический процессор	Mali400 с аппаратными 3D-ускорением и декодированием видео
ОЗУ	DDR3 512 МБ / 1 ГБ, ~100 МБ резервировано для GPU
Последовательный порт	3.3v TTL 4-контактный разъем
Вход аудио	3,5-мм микрофонное гнездо
Выход аудио	Через HDMI
USB	2×USB A 2.0
Внутренняя память	4 ГБ NAND, на Android доступно 1,5 ГБ в разделе пользователя
Внешнее хранилище	Слот для карты SDHC до 32 ГБ
Сеть	10/100 Ethernet, Realtek 802.11n Wi-Fi
Загрузка	С SD-карты или внешнего хранилища через u-boot



➤ Отладочная плата Cubieboard.

На плате есть Ethernet, что является огромным бонусом. Огорчало только отсутствие SATA-порта; и тем не менее за \$65 это очень неплохое решение для начинающего embedded-разработчика.

Спустя пару месяцев Miniland выпустила еще одну отладочную плату, под названием Cubieboard — [https://www.miniland.com/products/Cubieboard %20Developer %20Board](https://www.miniland.com/products/Cubieboard%20Developer%20Board).

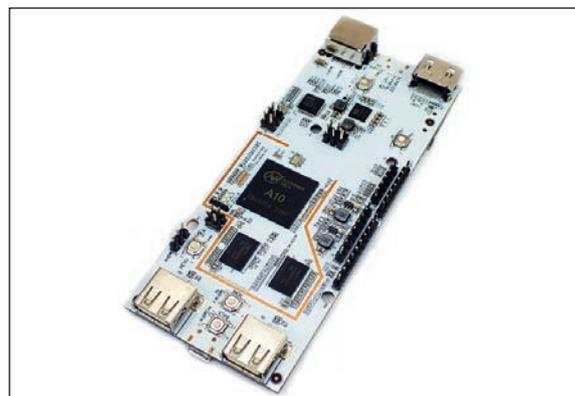
ЦПУ	1-ГБ ARM cortex-A8, NEON, VFPv3, 256-КБ кэш L2
Графический процессор	Mali400, OpenGL ES
ОЗУ	1-ГБ DDR3 480 МГц
Видеовыход	HDMI 1080p
Сеть	10/100M Ethernet, Wi-Fi опционально
Внешние накопители	4-ГБ NAND Flash
Ввод/Вывод	2×USB Host, 1×слот MicroSD, 1×SATA, 1×IR-сенсор
Расширенные интерфейсы	Расширенный 96-контактный интерфейс, включая I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP и др.
ПО	Доступна официальная версия Android (не предустановлен), Linux
Питание	Требует стабильного источника постоянного напряжения 5 В, 2 А со штекером 4,0-мм (внешний диаметр)×1,7 мм (внутренний диаметр)

Кроме Ethernet, SATA к плате приделал еще кучу разъемов, что предоставляло воистину огромный простор для всяческих модификаций, а цена в \$49 удельвала RPi просто вчистую.

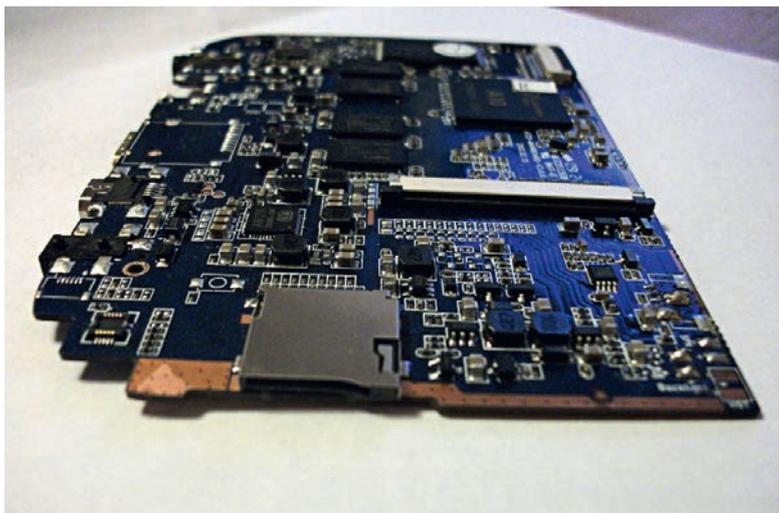
Недавно вышла еще одна интересная плата — pcDuino.

Вот ее спецификации:

ЦПУ (SoC)	AllWinner A10 ARM Cortex A8 1 ГГц
Графический процессор	Mali400, OpenGL ES GPU
ОЗУ	1-ГБ DDR3 480 МГц
Внутренняя память	2-ГБ Flash + слот для SD-карты до 32 ГБ
Видеовывод	HDMI 1080p
Сеть	10/100M Ethernet, Wi-Fi опционально через USB-брелок
Ввод/Вывод	2×USB 2.0 Host



➤ pcDuino отлично поладит с Arduino.



► Плата Gooseberry Board легко поддается разгону.

Сами разработчики как главное достоинство карты позиционируют то, что она совместима с Arduino. Вполне вероятно, что для многих Arduino-программистов это будет весьма солидным преимуществом.

А вот разработчики еще одной платы, Gooseberry Board, утверждают, что ее производительность втрое превосходит производительность Raspberry Pi!

ЦПУ	AllWinner A10 ARM Cortex A8 1 ГГц
Графический процессор	Mali-400, OpenGL ES GPU
ОЗУ	1-ГБ DDR3 480 МГц
Внутренняя память	4 ГБ + для карты microSD до 32 ГБ
Видеовывод	HDMI
Сеть	Wi-Fi 802.11 b/g/n
Порты	AC, 3,5-мм микрофонный разъем, Mini-USB
Совместимые ОС	Android ICS, Ubuntu*



► MarsBoard — то ли космическая, то ли воинственная...

Авторы пишут, что Ubuntu может быть установлена разве что при наличии некоторых продвинутых технических знаний, однако я так не считаю: процессор — тот же A10 Allwinner, который используется и на других устройствах, да и Ubuntu может быть установлена точно тем же способом, как и на остальные устройства. На мой взгляд, в том, чтобы использовать утилиту *dd* с SD-карты, нет ничего сложного, так что говорить про “advanced knowledge” — только отпугивать потенциальных членов сообщества и покупателей.

Цена, кстати, достаточно кусачая для такого устройства: примерно \$60.

И, наконец, погуляв по сетевым магазинам электроники, я наткнулся еще на штуковину под названием MarsBoard.

Судя по техническим характеристикам, девайс совсем не выглядит марсианским, потому что ничего экзотического в нем нет, совершенно стандартный набор.

ЦПУ	AllWinner A10 ARM Cortex A8
Графический процессор	Mali-400, OpenGL ES GPU
ОЗУ	1-ГБ DDR3 480 МГц
Внутренняя память	4-ГБ Flash + карта microSD до 32 ГБ
Видеовывод	HDMI
Сеть	10/100 М Ethernet
Ввод/Вывод	Коннектор SATA, 2×2.0-мм GP, 2×USB host + microUSB
Питание	5 В, 2 А

Стоит такое счастье всего \$49 вместе с доставкой, от других устройств на базе Allwinner оно не отличается — это опять же означает, что на данную плату можно поставить все дистрибутивы Linux, собранные под ARM-платформу, то есть Ubuntu, Debian, Fedora и пр.

Пока статья готовилась, меня спросили, какой бы компьютер посоветовал я. Я задумался и выдал, что если не отказывать себе в количестве средств, которые можно потратить на устройство подобного рода, то я бы взял что-то на процессорах Exynos, с 4 ядрами и SATA-портом. Однако если нужно какое-то бюджетное решение для тестирования встраиваемых систем, я бы однозначно выбрал Cubieboard. На мой взгляд, Cubieboard — отличный пример баланса цены, качества и превосходной функциональности; кроме того, у Cubieboard есть достаточно активное сообщество с живым списком рассылки.

На этом, пожалуй, все о самых ярких представителях одноплатных компьютеров. На самом деле их количество растет не по дням, а по часам. Желющие могут ознакомиться с характеристиками добрых четырех десятков таких — сравнительную таблицу на http://dl.dropboxusercontent.com/u/4035896/a320/downloads/SBC_comparison44.pdf любезно подготовил пользователь omgFIRE_id.

Бешеная популярность одноплатных компьютеров объясняется тем, что на их основе легко создать дешевые и компактные решения для бытового применения — например, для «умного дома». В данной статье я не рассматривал одноплатные компьютеры с точки зрения открытости исходного кода: это вопрос достаточно сложный и, вероятно, потребует отдельной статьи. **LXF**

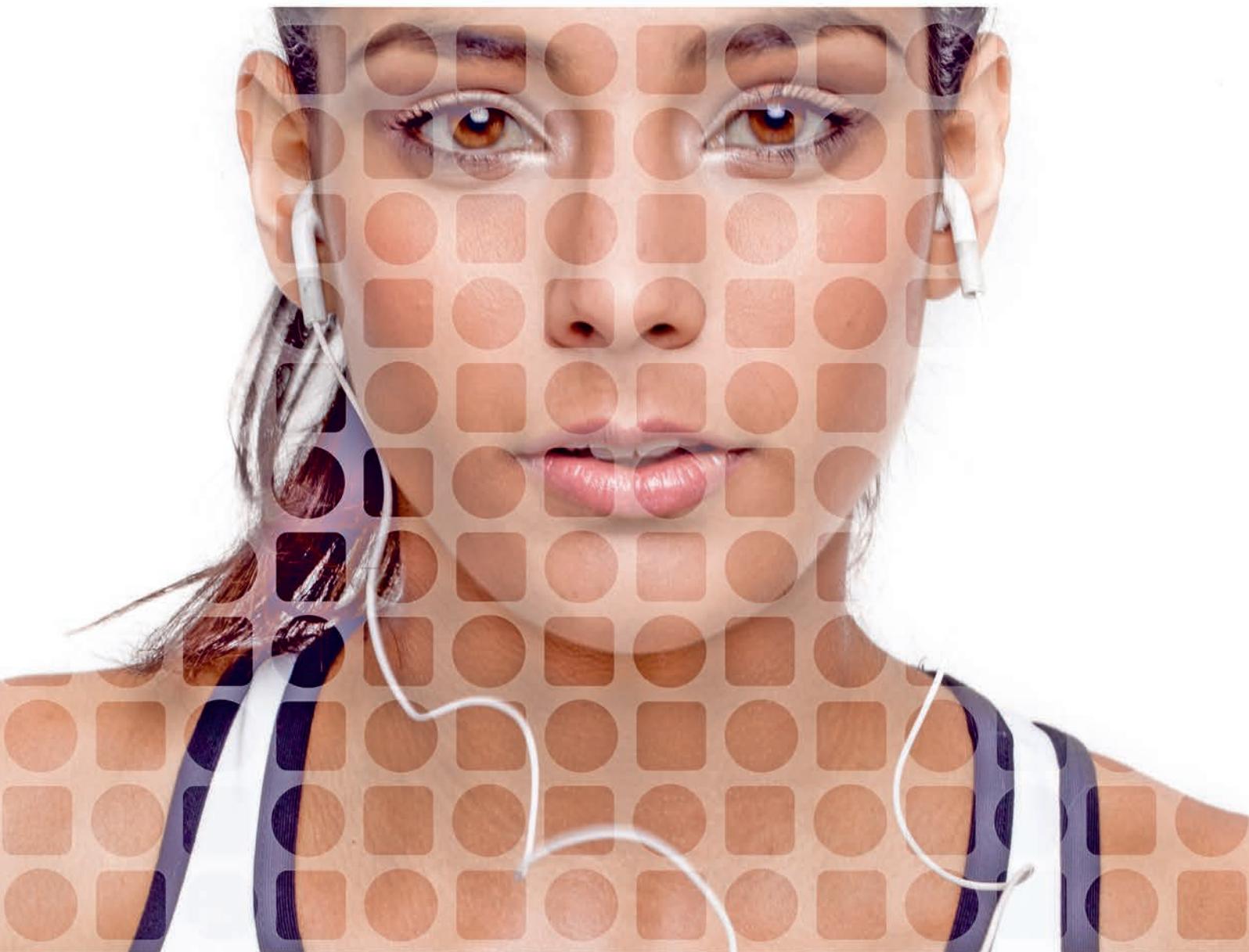
074

key.ru



KEY

техника и человек



ТВОИ ПОМОЩНИКИ В ТРЕНИРОВКАХ



Плееры



Smart-здоровье



Экшн-камеры



Весы



Фитнес-часы

Узнай подробнее на key.ru.

Товар сертифицирован. Количество товара ограничено.

Под сенью X.org

Последний рассказ Алексея Федорчука про историю Иксов — дальше уже начинается современность...

Прошлая статья (LXF169) завершилась тем, что на базе последнего кандидата в релизы XFree86 4.4 RC2 был создан его форк. А вслед за этим происходит событие, приведшее к Великому расколу. Точнее, послужившее его непосредственным поводом — причины его, как мы видели в предыдущей статье, были гораздо глубже.

Повод к Великому расколу

Что потом началось — не опишешь в словах...

Владимир Высоцкий

В феврале 2004 года выходит долгожданный релиз XFree86 4.4, аккумулирующий все новшества предшествующих корректирующих и «полуажурных» релизов. Однако — под скорректированной же лицензией. Если ранее XFree86 распространялась под стандартной «разрешительной» лицензией MIT, то в новой версии появился пункт, несколько напоминающий пресловутую «оговорку о рекламе» из первой версии BSD-лицензии, впоследствии изъятую (подробности см. в LXF146).

«Почему бы не помянуть добрым словом разработчиков?»

MIT и X

То, что в просторечии обычно называется лицензией MIT, правильней именовать лицензией X11. Ибо в разных проектах MIT использовалось несколько видов лицензий, большинство из которых со временем вышли из употребления. Существующий же текст лицензии, под которой распространялась XFree86 вплоть до версии 4.4, был сочинен специально на ранней стадии разработки Иксов и именно для них.

Хотя, если вчитаться в текст лицензии, выясняется, что все ее новшество сводится к требованию включать в документацию дистрибутивов, использующих XFree86, такую фразу:

«Данный продукт включает программное обеспечение, разработанное The XFree86 Project, Inc (<http://www.xfree86.org/>) и его сотрудниками».

Казалось бы — чего страшного? Почему бы лишний раз не помянуть добрым словом разработчиков хорошей (и на тот момент времени практически незаменимой) системы? А вот именно ее незаменимость и вызвала, думается, претензии: фактически каждый дистрибутив Linux общего назначения или операционная система BSD-семейства включали в себя Иксы — а альтернатив XFree86 к тому времени не было, ибо прочие X-сервера, о которых

вскользь упоминалось в одной из предыдущих статей (см. LXF168), практически вымерли. И такое требование было воспринято сообществом (сначала — некоторыми, но влиятельными его членами) как

беззастенчивая реклама проекта XFree86 за счет разработчиков всех остальных, взаимосвязанных, но независимых проектов.

Так что дальнейшие события можно описать цитатой из классика, чьи слова приведены и в качестве эпиграфа раздела: «Тут поднялся галдеж и лай...»

Новая лицензия была названа несовместимой с принципами свободы такими авторитетами, как Ричард Столлмен и разработчики Debian. К ним постепенно присоединились остальные ведущие разработчики, включая Тео де Раадта [Theo de Raadt] (OpenBSD) — его высказывания в этом плане были, пожалуй, наиболее резкими.

Правда, все эти разработчики поначалу пошли своими путями. В одни системы (Debian, OpenBSD) была включена XFree86 под последней «чистой» MIT-лицензией, большинство же переключилось на ее форк — X.org, первый официальный релиз которого

Хоть горшком назови

Новая лицензия проекта XFree86 по-научному называлась Лицензия XFree86 версии 1.1 (под версией 1.0 следует понимать лицензию X11). Интересно, что GNU/FSF, в свое время столь категорически ее осудившая, ныне признает ее совместимость с лицензией GPLv3 — той самой, которую FSF в настоящее время считает наиболее правильной и всячески рекомендует ее к употреблению. Однако с GPLv2 лицензия XFree86 версии 1.1 несовместима по-прежнему — из-за требования упоминания ее имени в документации (подробности здесь: <http://www.gnu.org/licenses/license-list.htm>).

Впрочем, ныне это уже не актуально. И может быть приведено только в качестве примера того, как юридические формулировки, придуманные, казалось бы, с самыми лучшими намерениями — ради защиты свободы — вступают в противоречие со здравым смыслом. А в конечном счете — и с той же самой свободой. Еще один такой пример мы уже видели при рассмотрении истории файловой системы ZFS (LXF164).

появился в апреле 2004 года. Который вскоре и стал магистральной линией развития Иксов.

Конечно, были и отдельные «голоса из ветвей». Так, Патрик Фолькердинг [Patrick Volkerding] высказался в том смысле, что его все эти политико-юридические игры не интересуют, и ничего крамольного в новой лицензии XFree86 он не видит. Однако был вынужден, в целях совместимости с прочими дистрибутивами Linux, перейти на X.org.

А вот разработчики NetBSD вообще сочли, что новая лицензия вполне нормальна с точки зрения BSD-стиля, и использовали XFree86 вплоть до версии 4.0 включительно (вышла в самом конце 2007 года). Правда, в версии 5.0 (апрель 2009) и им пришлось от нее отказаться, ибо разработка XFree86 к этому времени фактически прекратилась: последний релиз, 4.8, появился в конце 2008 года.

Конец XFree86...

That is the end of Solomon Grandy.
Английское народное

Причин прекращения разработки XFree86 было несколько. Здесь можно назвать и ее невестребованность в связи с переходом большинства, а затем и всех дистрибутивов Linux и BSD-систем на X.org, и миграцию многих, если не большинства, бывших ее разработчиков в том же направлении. Но главной, на мой взгляд, причиной было то, что в XFree86, в сущности, стало нечего разрабатывать: уже к 2004 году она представляла собой устоявшуюся экосистему, кардинальные улучшения в которую можно было



» Хотя Ричард Сталлмен весьма ревниво настаивает на префиксе GNU, лицензию XFree86 он осудил.

внести только путем столь же кардинальных изменений. Так что участникам проекта, в сущности, оставалось только отлавливать баги и вносить коррективы в соответствии с изменениями остальных базовых компонентов. То есть осуществлять банальную техническую поддержку конечного продукта. А это для разработчиков Open Source, следующих курсу Just for Fun, что серпом по... ушам. Вот хлопцы и разбежались в разные стороны. Точнее, в сторону X.org.

Однако в последнем случае нельзя исключить и влияние активной пропаганды порочности новой лицензии XFree86 с точки зрения идеалов свободы и демократии. А насколько такого

рода пропаганда может быть действенной в сообществе Open Source, мы имели возможность убедиться совсем недавно — на примере тотальной systemd'изации всего Linux'a. Так что уделим обсуж-

дению этого вопроса еще несколько минут.

Как уже было сказано, с точки зрения здравого смысла и человеческой порядочности новая формулировка лицензии XFree86 не может вызвать никаких возражений: упоминание автора всегда считалось хорошим тоном в мире фундаментальной науки. А разработка Open Source, как еще давно показал Николай Безруков, есть разновидность фундаментальной науки. По крайней мере, должна ею быть...

Интересно, что наиболее активные из тогдашних критиков новой лицензии XFree86 (не будем лишней раз поминать их все) и по сей день не упустят случая напомнить, что ОС Linux на самом деле должна называться GNU/Linux. Ибо Linux — это только ядро, а все пользовательское окружение — достижения проекта GNU.

Но, если следовать этой логике, то термин вроде XFree86/Linux имел 10 лет назад куда больше прав на существование. Как сейчас более правомерен был бы термин X.org/Linux. А во все времена и у всех народов — просто X/Linux. Ибо, с одной стороны, так называемое пользовательское окружение ядра Linux состоит не только из GNU-компонентов. Более того, существуют

«Нельзя исключить и влияние активной пропаганды.»

Считалочка

Дабы отрешиться от старого мира, в проекте Xorg отказались от продолжения нумерации версий XFree86. И для начала продолжили нумерацию спецификаций оконной системы X вообще: первый релиз проекта (апрель 2004) именовался просто X11R6.7.0. Напомню, что предыдущая «общеейсовая» версия, X11R6.6, появилась на свет в апреле

2001 года. А спецификации «мажорные» X11R6, на протяжении многих лет лежащие в основе XFree86, уходят в далекий майский день 1994 года.

Со временем параллельно ей стала использоваться нумерация по версиям собственно сервера X.org. В релизах X11R6.7.0 и X11R6.8.X в неявном виде

подразумевалось, что они имеют номер версии 1.0. А далее к нему приписывалась «мажорная» (1.X) или «минорная» (1.X.Y) приставка.

В настоящее время именование по версиям сервера X.org является основным. Так, текущая его версия на момент сочинения этих строк — 1.14.1.

Вам срочный пакет

Традиционно исходные коды XFree86 распространялись в виде нескольких крупных tar-архивов (в разных версиях — от трех до семи). Которые при монолитной системе сборки и в двоичном виде собирались как серия крупных пакетов, таких, как Xbin, Xlib, Xxserv и так далее. Разумеется, бинарники можно было собрать и более мелко, и мейнтейнеры ряда дистрибутивов, таких, как RedHat и Debian, прибегали к этому со стародавних времен. Но поначалу в Xorg

штатно такая возможность не использовалась. Как не применялось «дробное» пакетирование и в дистрибутивах, придерживавшихся соответствия пакетов «авторских» и «дистрибутивных» — а такими на протяжении долгого времени были Slackware, Gentoo и идеологически близкие к ним, не говоря уж об LFS.

С переходом к модульной системе сборки бинарников X.org как огромного множества небольших пакетов стала нор-

мой во всех дистрибутивах. Более того, и сами исходники Xorg стали распространяться в виде изобилия мелких тарболов «целевого назначения», что обеспечивало существенно большую гибкость при построении конечной системы.

Впрочем, чтобы, как говорится, почувствовать разницу, достаточно сравнить древа исходных текстов версий X11R6.9 и X11R7.0 на официальном сервере проекта — ftp.x.org/pub.

дистрибутивы (например, Tiny Core), где их нет совсем. А вот без Иксов в том или ином их проявлении ни один дистрибутив общего назначения не обходился никогда.

Со стороны же другой, современный пользователь Linux может (то есть имеет не только право, но и все шансы) даже не подозревать о пресловутом пользовательском GNU-окружении. Ибо часто работает исключительно в графической среде, то есть в Иксах и надстраивающих их оконных менеджерах или интегрированных десктопах. А потому без XFree86/X.org ни о каком десктопном Linux'е не может идти и речи.

Так что новая лицензия XFree86 была только попыткой «легализовать» существующее положение дел. Причем без претензий на замену устоявшихся имен операционных систем, а только на уровне обязательности упоминания одного из важнейших, с точки зрения пользователя, их компонентов. И более того, не во всяких там копиях и копилейтах, а исключительно в документации, сопровождающей дистрибутивы. Тем не менее, она была дружно признана несовместимой с духом изначальных Иксов (Тео де Раадт) и свободного софта вообще (Ричард Столлмен) и с буквой «самой свободной» из всех свободных лицензий, то есть GPLv2. И потому подверглась всеобщему остракизму.

... и начало X.org

Если историю XFree86 уже в 2004 году можно было считать законченной (хотя она существует и доступна по сей день), то история X.org только начиналась. И начиналась она весьма бурно.

Итак, в апреле 2004 года появляется первая версия собственной X.org — X11R6.7.0, основанная на исходниках XFree86 4.4 RC2 и мало чем от последней отличающаяся. А далее версии сменяются с быстротой, заставляющей вспомнить ранние времена первоначальных Иксов (**LXF168**): 8 сентября 2004 — X11R6.8.0, 17 сентября 2004 — X11R6.8.1, февраль 2005 — X11R6.8.2.

В версии X11R6.8.0 впервые появляются такие ныне привычные вещи, как Composite — предварительная прорисовка изображения для вывода его на экран в уже готовом виде, прозрачность

окон и прямая поддержка конфигураций с несколькими мониторами. «Минорные» же версии носили корректирующий характер.

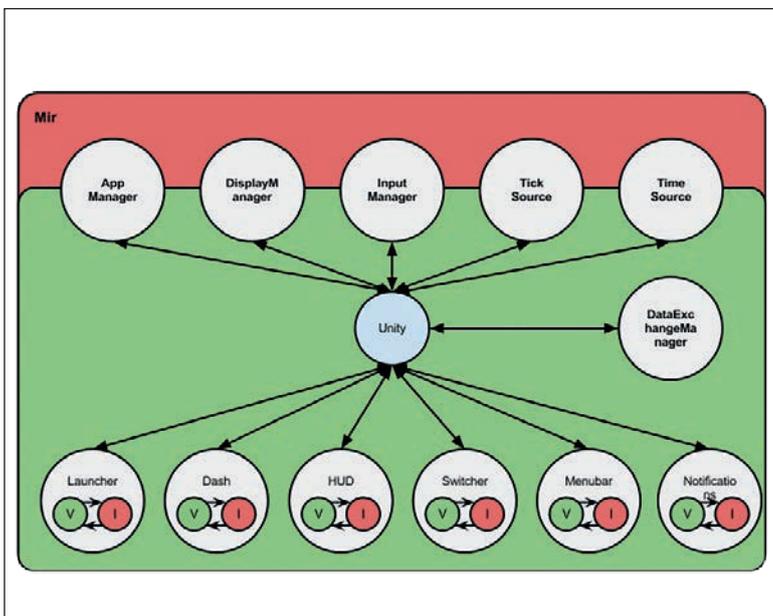
Далее наступает некоторое затишье, продолжавшееся

«Близится конец Иксов: на горизонте маячат Wayland и Mir.»

до 21 декабря 2005 г. Зато этот день ознаменовался выходом сразу двух версий — X11R6.9 и X11R7.0. Нет, это было не раздвоением личности, а окончательным разрывом со старыми традициями: переходом от Imake — системы автоматизации сборки, унаследованной от XFree86 — к Autotools, аналогичной системе, развиваемой в рамках проекта GNU. Что вызвало и переход от монолитной сборки к модульной. В результате чего на одной кодовой базе и были созданы монолитная версия 6.9 и модульная версия 7.0. Во всех последующих релизах X.org использовалась только система Autotools, и, соответственно, все они были модульными.

Следующая веха в истории X.org — версия X11R7.3 (X-сервер 1.4), вышедшая в сентябре 2007 года: в ней, среди прочего, получает дальнейшее развитие автоопределение оборудования, в том числе и горячего подключения. Тогда это делалось через HAL (Hardware Abstraction Layer) — и делалось вполне успешно; и, что характерно, в сборках Иксов не только для Linux, но и для BSD-систем. В частности, автор этих строк неоднократно использовал HAL freeBSD.

Тем не менее, в версии X11R7.6 (X-сервер 1.8.0, декабрь 2010) в управлении устройствами подсистема HAL была заменена менеджером устройств *udev*. Что, с одной стороны, привело к определенному прогрессу в этом деле. А с другой, поскольку *udev* — инструмент, специфический для Linux'a, отгородило последующие версии X.org от остальных UNIX-подобных систем. Но на этом я поставлю точку: мой рассказ подошел к своему логическому завершению. И вместе с ним близится и конец Иксов вообще: на горизонте маячат Wayland, с одной стороны, и Mir — с другой. Но это уже дела дней сегодняшних и грядущих, а не предмет истории. **LXF**



➤ Железный конь идет на смену крестьянской лошадке? Ubuntu провозглашает Mir...

23-25 октября, 2013 г. Москва. Центр Digital October

Приглашаем

Аудитория:
до 800 делегатов
из 250+ компаний
и 20+ стран

Приз 1000 Евро
за лучший
исследовательский
доклад

Бесплатное
участие для
докладчиков

- Программистов и инженеров качества
- Системных аналитиков и архитекторов
- Лидеров команд и менеджеров проектов
- HR-специалистов и руководителей производства
- Исследователей, студентов, аспирантов
- Продуктовые компании, компании, реализующие ПО «в облаке»
- Центры разработки и филиалы транснациональных компаний
- Аутсорсинговые компании и ИТ-департаменты государственных учреждений

Подробнее: www.secr.ru, contact@secr.ru, +7 812 336 93 44



Спонсоры



Партнеры





Что за штука...

Tizen

Марко Фиоретти представляет очередную ОС на базе Linux, готовую вселиться в первый же подходящий смартфон.

В Ну, рассказывайте про этот Tizen. Как-то так называлась красная газировка, популярная у детишек-сладкоежек в 1980-е...

О Tizen (www.tizen.org) — это открытая операционная система и программная платформа для любого типа компьютеров — но в основном для мобильных устройств.

В А как следует понимать «в основном для мобильных»?

О Что Tizen предлагает окружение и приложения для нетбуков, смарт-ТВ и автомобилей.

В Еще и автомобилей?.. В таком случае, как это работает?

О Уже идет работа над платформой Tizen In-Vehicle Infotainment (<https://wiki.tizen.org/wiki/IVI>), с которой приложения Tizen станут доступны в «автомобилях, автобусах и самолетах».

В Кто же тогда стоит за Tizen, и кто ее контролирует?

О Tizen — это проект Linux Foundation. В сфере деятельности Tizen Association входит маркетинг и обучение, а также выработка требований и другие функции; но важные решения принимает только Technical Steering Group. На данный момент, самыми крупными и активными членами являются Intel и Samsung. Помимо них, Tizen в той или иной степени взаимодействует с такими произво-

дителями, как Panasonic и Fujitsu, и мобильными операторами NTT Docomo, Sprint и Vodafone.

В Tizen ведь не первая платформа типа «Linux для мобильных», даже без Android, верно?

О Верно. Не вдаваясь в подробности, Tizen является или претендует на то, чтобы быть последователем (по крайней мере) Maemo, Moblin, LiMo Foundation и платформы Bada/Wave от Samsung.

В То есть телефоны на Wave можно будет обновить до Tizen?

О Э-э, ну... нет. Приложения Bada будут портированы на Tizen, а в Samsung заявили, что

владельцам телефонов Wave придется «делать выбор в пользу лучшего», что на языке маркетинга означает «нет уж, покупайте телефон Tizen».

В Очень мило. Но оставим Bada; нужна ли нам вообще еще одна мобильная ОС на Linux?

О Разнообразие — это хорошо, разве не так любят говорить в свободном ПО? Но, помимо принципов, есть и более практическое объяснение. До прошлого года на рынке смартфонов было всего четыре ключевых игрока (поставщиков как оборудования, так и ПО): Apple, создающий и то, и другое самостоятельно; Nokia и Microsoft; Google и HTC; а также Samsung, не менее преуспевающий, но только за счет Android.

Насколько я понимаю (это моя личная теория), ограничивать Samsung в правах на использование Android — не в интересах Google: учитывая успех линейки Galaxy, сейчас было бы безумием от него отказываться.

И все же, при текущем положении вещей, столь сильная зависимость Samsung от Android отнюдь не радует его руководителей, так? Для Samsung, Tizen — своего рода страховочная мера, дополнительный аргумент в переговорах с Google, без «капитуляции», так сказать, в пользу Microsoft. В отличие от Nokia и HTC, Samsung достаточно велик и могуч, чтобы привести Tizen к успеху, по крайней мере на рынке бюджетных телефонов. Ну или, по крайней мере, они сами так считают.

«Для Samsung, Tizen — страховочная мера, лишний аргумент в переговорах с Google.»

В Верна ваша теория или нет, но факт тот, что судят смартфон по его приложениям. Их в Tizen не меньше, чем в Android? Или, даже лучше, можно ли будет запускать приложения Android на Tizen?

О На самом деле, помимо имеющегося ПО Bada, плюс нового HTML5 и портирования любых других программ, у вас будет по крайней мере один способ запустить оригинальные приложения Android на Tizen...

В Звучит intriguing. С этого места, пожалуйста, подробнее...

О Это все благодаря Android Application Compatibility Layer (ACL, <http://openmobilew.com/products.php>) из OpenMobile. Утверждается, что его библиотеки гарантируют 100% совместимость для всех приложений Android на многих операционных системах и устройствах — от смартфонов до автомобильных аудиосистем — без потерь производительности.

В Ну, «без потерь» — это, по-моему, слишком громко сказано, но меня больше интересует другое. Могу я установить этот ACL самостоятельно?

О Да. OpenMobile взаимодействует непосредственно с OEM (Original Equipment Manufacturers [производителями оборудования]) и мобильными операторами, и их основная задача — сделать так, чтобы телефоны были оснащены ACL еще до момента их покупки. При этом OpenMobile все же предлагает свободное сетевое приложение ACL, Интернет-магазин AppMall для клиентов ACL, а также специальный обозреватель к нему. AppMall включает 10 хранилищ внутри одного каталога из более чем 250 000 приложений Android.

В Это обнадеживает, по крайней мере, пока. Хорошо, не могли бы вы тогда рассказать побольше о том, как работает Tizen?

О Можно сказать, что Tizen основывается на двух принципах: первый состоит в том, что приложения Android можно рекомпилировать или просто использовать с ACL; второй — что «будущее» — за приложениями на основе HTML-5», соответственно с этим набором технологий и должен быть оптимизирован Tizen.

В Но в Tizen все равно используется ядро Linux, не так ли?

О Правильно. В основе стека Tizen — модифицированное ядро Linux и совокупность драйверов устройств. Далее, снизу вверх, следуют: Основная подсистема [Core], Системное окружение [Native framework] и Сетевое окружение [Web

framework]. Подсистема ядра состоит из базовых открытых библиотек и API.

В А за что несет ответственность Системное окружение?

О Это совокупность системных сервисов и библиотек, объединенных в пространства имен. Отдельно для графики, поддержки локализаций, безопасности и прочего. В совокупности, они составляют тысячи открытых интерфейсов, при помощи которых можно создавать системные приложения.

В В чем же тогда функции Сетевого окружения?

О Еще одной частью мозаики является, так или иначе, поддержка HTML5 и интеграция со всеми базовыми сервисами, от аудио и видео, до геолокации и сообщений.

В Звучит замечательно, но насколько это уже воплощено?

О Первая стабильная версия появилась в 2012. Помимо всего прочего, она включала WebRTC, систему общения в реальном времени с помощью браузера (которую мы обсуждали

в одной из прошлых статей *Что за штука?*) и графические библиотеки EFL из *Enlightenment*.

В Я так понимаю, сейчас речь идет об еще одной стабильной версии?

О Верно. Tizen 2.0 — также именуемая Magpollia — вышла в начале 2013. В ней появилось несколько новых системных приложений и новые API устройств, с поддержкой Bluetooth и NFC.

Разработчики получили прямой доступ к истории звонков, подсистемам календаря и сообщений. На Tizen 2 запускаются как собственные приложения, так и приложения HTML5, а также те, которые могут использовать оба интерфейса. Также реализована поддержка преобразования текста в речь и фоновой работы программ.

В А как насчет Qt? Его нам светит увидеть в Tizen?

О Насколько мне известно, не по умолчанию. Однако некоторые сейчас работают над портированием Qt на Tizen, или, по крайней мере, в этом заинтересованы.

В Что же мне понадобится, чтобы начать создавать приложения Tizen для Linux?

О Для начала (по крайней мере, на март 2013), нужен компьютер с Ubuntu 11.10 или 12.04 и Oracle Java (не OpenJDK) версии 6 или выше. Затем потребуется скачать набор для разработки и исходный код с <https://developer.tizen.org/>. То есть потребуется правильное оборудование.

В Правильное? А в чем должна выражаться его правильность?

О Это значит, что для максимально быстрого и реалистичного тестирования ваших приложений вам понадобятся графическая карта и веб-камера, которые фигурируют в списке поддерживаемых на сайте Tizen. Да, и еще нужен процессор Intel с поддержкой Intel VT, чтобы использовать HAXM.

В HAXM что?..

О Hardware Accelerated eXecution Manager — это аппаратный движок виртуализации, ускоряющий работу эмулятора Tizen, в составе набора для разработки. Но только — как я уже говорил — на правильном процессоре.

В Это окружение работает только из командной строки?

О Нет. Существует системное графическое окружение WYSIWIG, с графическим инструментом тестирования, менеджером проектов, просмотром журналов и другими радостями.

В Понятно. И последний вопрос: а вы бы купили себе телефон на Tizen?

О Если это значит, что я стану владельцем открытого телефона, который я могу настраивать, как хочу, то конечно! Если нет — я не знаю. Но это, конечно, только мое личное мнение. **LXF**



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.

Сдача позиций? Я так не думаю!

Некоторое время назад Ubuntu покинул верхние строчки рейтингов дистрибутивов и получил свою дозу критики (среди прочего) за рабочий стол Unity. Некоторые могут сказать, что Canonical (компания-разработчик Ubuntu) сдает позиции. Я не согласен.

На рынке настольных систем HP выпустила новый ПК «все в одном» с Ubuntu за 349 фунтов. Ubuntu Touch набирает разгон и уже используется на некоторых сенсорных устройствах Nexus — у меня было одно такое — хотя еще не является законченным продуктом (подробности на omg.ubuntu.co.uk).

Конечно, Ubuntu сделал себе имя как настольная система, но Canonical активно прорывается на рынок серверных систем, и за несколько лет у них было несколько серверных релизов. Кроме того, Ubuntu также становится невероятно популярным в качестве образа в облаке. Зайдите в мастер запуска Amazon Web Services и запустите образ EC2, и вы увидите, что четыре из двадцати образов для быстрого запуска — это Ubuntu. Взгляните на образы, предоставленные сообществом (их 15596!), и расклад получится примерно такой: Ubuntu — 31,8 %, CentOS — 3,5 %, Red Hat — 2,5 %, SUSE — 0,3 %, Windows — 9,8 % и прочий Linux — 48,3 %. Особенно интересно выглядит соотношение Ubuntu/Red Hat.

Еще один растущий рынок — частные облака. Canonical агрессивно наступает и здесь, предлагая инновационные решения для развертывания, такие как MAAS и Juju, которые могут революционизировать способ не только развертывания компьютеров на облаках, но и самого построения облаков.

Кажется, у Canonical разработаны стратегии по всем направлениям технического прогресса, хотя им еще предстоит меня убедить в целесообразности перехода на возобновляемые релизы вместо традиционного релиза раз в полгода.

chris.linuxformat@gmail.com

Эзотерическое системное администрирование из причудливых заворотов кишок серверной



Металл как сервис

MAAS в Ubuntu Server 12.04 расширяет концепцию услуг по требованию на мир «железа».

Одна из определяющих характеристик облачных вычислений — быстрое предоставление услуг, что позволяет запускать и создавать образы виртуальных машин (VM), а затем уничтожать их по желанию. За всем этим, конечно, стоит настоящее «железо». Пусть у вас есть дата-центр. Стойки забиты компьютерами, и каждую неделю добавляются новые. Каждый компьютер нужно физически установить и подключить, затем поставить на него ОС, а также все необходимые для его работы программы. Для переноса компьютера в другой проект впоследствии потребуются ручное вмешательство. MAAS предлагает альтернативу: расширить перечень требований к VM облака до физических компьютеров.

MAAS содержит сервер услуг, который предлагает сервисы TFTP и DHCP для ответа на запросы загрузки нового оборудования по PXE. Вот что

происходит при запуске только что подключенного сервера:

- 1 Сервер отправляет загрузочный запрос PXE.
- 2 и 3 Серверы DHCP и TFTP в MAAS отвечают на запрос и устанавливают на компьютер небольшой ISO-образ.
- 4 Этот образ создает перечень оборудования компьютера и отправляет его обратно MAAS, который записывает его в базе данных.
- 5 Компьютер выключается.

В этом пункте компьютер входит в состояние готовности, то есть он готов к повторному пробуждению. Затем в дело вступит утилита развертывания Juju от Canonical. Она свяжет компьютер с проектом и установит на него необходимое ПО.

У MAAS приятный web-интерфейс — им я и советую пользоваться, так как клиент командной строки неудобен. В MAAS также можно управлять VM и строить облако внутри облака; это удобный и простой способ знакомиться с системой, если у вас нет доступа к запасной стойке с шасси.

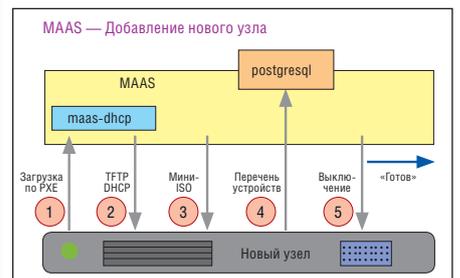
Расширяем жаргон

PXE Pre-Boot Execution Environment — предзагрузочная среда выполнения (позволяет компьютеру загружаться с сервера по сети).

DHCP Dynamic Host Configuration Protocol — протокол динамической настройки хоста (используется для динамического назначения IP-адресов компьютерам).

TFTP Trivial File Transfer Protocol — простой протокол передачи файлов (используется для загрузки образа ядра на узел).

ISO Файл, содержащий образ CD.



➤ Новые компьютеры добавляются в MAAS во время первоначальной загрузки по PXE.

Коротко о Tomcat

Часть 2 Осваиваем конфигурацию сервера Tomcat и узнаем, что иногда XML не избежать.

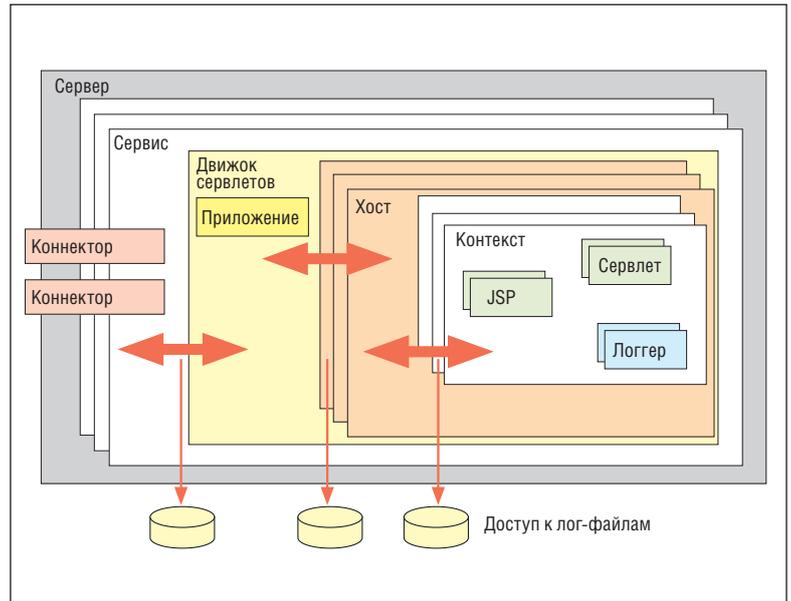
В прошлом месяце мы установили *Tomcat*, написали сервлет и развернули его. Как мы узнали, у *Tomcat* сразу есть вполне рабочие настройки. Но ведь ни одного уважающего себя системного администратора не удовлетворяют настройки по умолчанию? Поэтому на нашем уроке мы немного поработаем с файлом настройки *Tomcat* `/etc/tomcat6/server.xml`. (Если вы со мной с самого начала, то помните, что я пользуюсь стандартными пакетами *Tomcat* из репозитория CentOS, установленными в CentOS 6.2.) Немного сокращенная версия этого файла показана ниже. Я только убрал комментарии и несколько более продвинутых настроек, но файл по-прежнему вполне рабочий. Компоненты этого файла имеют строгую иерархическую структуру, которой точно соответствуют вложенные прямоугольники, которые я нарисовал на рис. 1:

```
<Server port="8005" shutdown="SHUTDOWN">
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3"
      redirectPort="8443" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm
        UserDatabaseRealm"
        resourceName="UserDatabase"/>
      <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true"
        xmlValidation="false" xmlNamespaceAware="false">
        </Host>
      </Engine>
    </Service>
  </Server>
```

Рассмотрим некоторые компоненты более подробно. Коннектор (Connector) — компонент, который слушает соединения на указанный порт. В нашем файле настройки два коннектора. Первый — коннектор HTTP; он слушает порт 8080. Благодаря этому коннектору *Tomcat* может работать как самостоятельный web-сервер. Статический контент (включая чистый HTML и изображения) обслуживает напрямую, а другие запросы передаются движку сервлетов. Второй, коннектор AJP, слушает порт 8009, который можно выбрать, если вы хотите скрыть *Tomcat* за *Apache*. Подробнее об этом поговорим немного позже.

Движок сервлетов [Engine] представляет механизм обработки запросов, связанный с сервисом. Он принимает и обрабатывает запросы от одного или нескольких коннекторов и возвращает результат браузеру через те же коннекторы. Важный атрибут тэга `<Engine>` — `defaultHost`. Он определяет хост, который будет обрабатывать запросы, направленные хостам, обслуживаемым этим движком и не соответствующим ни одному из заданных хостов. (Вскоре, надеюсь, станет понятнее.)

Хост [Host] задает виртуальный хост (его можно представить как web-сайт), запросы для которого обрабатывает его движок. В *Tomcat* для реализации виртуального хостинга (то есть когда один сервер обслуживает несколько сайтов) достаточно поместить несколько контейнеров `<Host>` внутри одного движка Engine. Важные атрибуты тэга `<Host>` — `name` и `appBase`. Вместе эти атрибуты связывают имя сайта — например, `www.linuxformat.`



► Рис. 1. Иерархия конфигурации *Tomcat* прямо отражена в структуре его файла настройки XML.

`com` — с каталогом верхнего уровня, в котором находятся контент сайта и сервлеты.

Если вы знакомы с *Apache*, то это похоже на директиву контейнера `<VirtualHost>` в файле настройки *Apache*. Как и с виртуальным хостингом в *Apache*, нам потребуется небольшая помощь от DNS, который нужно настроить для разрешения имен наших сайтов на IP-адреса нашего сервера *Tomcat*. Позже мы создадим дополнительный виртуальный хост в нашей конфигурации *Tomcat*.

Осталось обсудить еще один компонент — `<Context>`. Он задает свойства web-приложения, запущенного на связанном с ним хосте `<Host>`. Компоненты `<Context>` можно размещать в тэгах `<Host>` прямо в `server.xml`, но в поставке *Tomcat* по умолчанию делается иначе. Вместо этого контексты определяются в каталогах вида `$CATALINA_HOME/conf/<engine_name>/<host_name>`. В нашей установке по умолчанию имя движка — `Catalina`, а единственный хост — `localhost`, поэтому имя файла будет таким: `/etc/tomcat6/Catalina/localhost`.

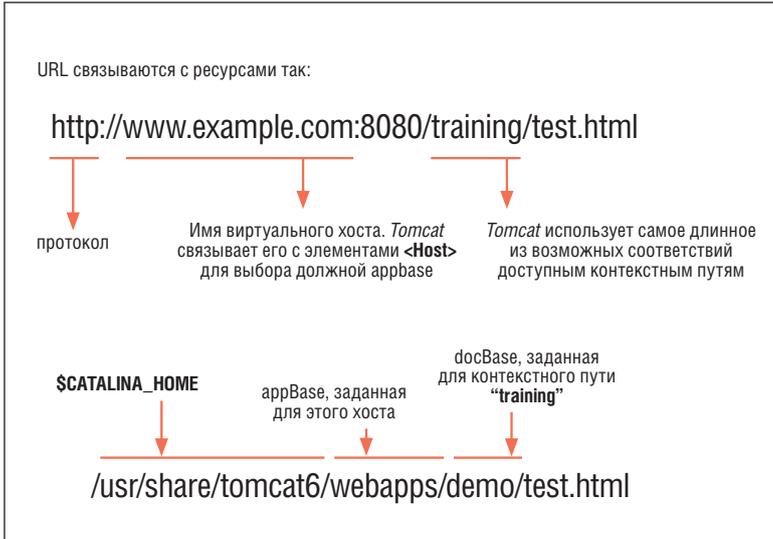
Контекст определяет приложение, выполняющееся внутри хоста. Например, он может выглядеть так:

»

Зоопарк O'Reilly

Некоторые из книг издательства O'Reilly лучше известны по животным на обложках. Классическое «Программирование на Perl», например, повсеместно называют «книгой с верблюдом». Иногда в них есть смысл: на обложке «Python в двух словах» нарисован питон, на обложке «Squid: исчерпывающее руководство» сидит, да,

кальмар, а на обложке «Безопасной оболочки» — улитка. Но почему на «Tomcat: исчерпывающее руководство» — снежный барс? Похоже, кот был уже занят — правда, не знаю, на какой книге он был. Если к вам придет озарение, напишите мне на `chris.linuxformat@gmail.com`. С удовольствием прочту ваши письма.



► Рис. 2. И имя хоста, и контекстный путь играют роль в определении ресурса, который будет обслуживаться. Этот рисунок якобы должен вам помочь.

```
<Realm className="org.apache.catalina.realm"
reloadable="true">
```

Путь контекста ("/training" на рис. 2 сверху) задает URL (часть, которая идет сразу после имени хоста), а *docBase* — имя каталога (в *appBase* для этого хоста), где находится контент соответствующего сайта.

Запутались? Нас таких много! Взгляните на рис. 2 и посмотрите, как URL преобразуется в имя ресурса, который на самом деле выдается по запросу.

Вентили управляют потоком

В эту базовую структуру сервера можно добавлять т. н. вентили — компоненты, которые отслеживают или фильтруют поток информации на различные компоненты сервера. Вентили доступа к лог-файлам, например, записывают в лог-файл запросы, которые обрабатывает сервер, и здесь удобно то, что область действия запросов, записываемых в лог, зависит от уровня, на котором они находятся в иерархии компонентов. Например, вентиль доступа к лог-файлу, помещенный в движок, запишет все запросы к этому движку. Вентиль логирования, помещенный в хост, запишет все запросы к этому хосту, а (ну вы поняли идею) вентиль логирования, помещенный в контекст, запишет только запросы к данному web-приложению. Вентили логирования показаны красными стрелками.

Давайте рассмотрим несложный пример вентиль для доступа к лог-файлу:

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
prefix="localhost_access_log."
suffix=".txt">
```

Шестерка и семерка

На нашем уроке мы пользуемся *Tomcat 6*, потому что эта версия находится в репозиториях CentOS, но сейчас все силы брошены на *Tomcat 7*, разрабатываемую с середины 2010 года. По сравнению с 6-й версией, в 7-й реализованы более свежие версии спецификаций сервлета и JSP. Также большее внимание уделено обнаружению и предотвращению утечек памяти (*Tomcat* долго ругали за утечки памяти при перезагрузке web-приложений). Есть поддержка включения внешнего контента прямо в web-приложение, и повышена безопасность приложений *Manager* и *Host Manager*.

```
pattern="common"
resolveHosts="false"/>
```

Атрибуты *prefix* [префикс] и *suffix* [суффикс] задают начальную и конечную части имени лог-файла. Для получения полного имени файла *Tomcat* помещает между ними дату. Поэтому полное имя может быть, например, таким: `localhost_access_log.2013-03-18.txt`. Атрибут *pattern* [шаблон] задает структуру данных, записываемых в лог-файл. Шаблон "common" — это общий формат записи, который также используется в *Apache* и многих других web-серверах. Это шаблон по умолчанию, и существует множество сторонних утилит для анализа и генерации отчетов по лог-файлам, записанным в этом формате. Формат записи можно указать и детально, с помощью обозначений, называемых идентификаторами шаблонов. Их довольно много, но чтобы вы просто поняли идею, взгляните на строку

```
pattern="%a %t"
```

Она запишет в лог IP-адрес клиента, а также дату и время.

Если вы еще настроены следовать за мной, добавьте вентиль доступа к лог-файлу в `/etc/tomcat6/server.xml`. Это довольно легко, так как там уже есть один — он просто закомментирован (чтобы найти его, просто поищите *AccessLog Valve*). Обратите внимание на расположение вентиль. Он находится внутри одного и единственного хоста `<Host>` с именем "localhost", поэтому будет записывать в лог только обращения к этому сайту. Удалите символы комментария, сохраните файл и перезапустите сервер:

```
# service tomcat6 restart
```

Теперь проверьте, существует ли лог-файл. Он должен быть в каталоге `/var/log/tomcat6`. Изначально он будет пустым, поэтому воспользуйтесь обычным приемом сисадмина и запустите на нем команду "tail -f", чтобы просматривать его содержимое в реальном времени. Попробуйте зайти на сайт (сгодится любая страница — хоть главная, хоть с документацией) и проверьте, появляются ли в лог-файле строки. Вот пример того, что можно увидеть. Эта строка также иллюстрирует общий формат записи лог-файла:

```
0:0:0:0:0:0:1 - - [23/Mar/2013:20:47:08 +0000] "GET /RELEASE-
NOTES.txt HTTP/1.1" 200 8669
```

Обратите внимание на адрес обратной петли в формате IPv6 в первом поле. (Сейчас Linux предпочитает по возможности разрешать имена в адреса IPv6, а не в IPv4.) Вы также увидите временную отметку запроса, первую строку запроса (из которой можно понять, какой ресурс запрашивался), код ответа HTTP (200 означает «успешно») и число байт, возвращенных в коде ответа HTTP, которое в данном случае соответствует размеру файла `RELEASE-NOTES.txt`.

Отключаем воду

С помощью вентилей также можно управлять доступом. Например, вентиль фильтра удаленного адреса позволяет управлять доступом на основе IP-адреса клиента. Для публичного web-сервера это не слишком полезно, потому что IP-адрес клиента ничего не говорит о том, что представляет собой клиент. Но это может пригодиться, если вы хотите ограничить доступ к приложению, заданным внутренним IP-сетям.

Вот пример фильтра удаленного адреса, который в этом случае разрешает доступ только с локального компьютера:

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="127.0.0.1" />
```

Также существуют вентили фильтра удаленного хоста, ограничивающие доступ на основе имени хоста клиента. Например:

```
<Valve className="org.apache.catalina.valves.RemoteHostValve"
allow="goodguys.com" />
```

Но учтите, что такой фильтр заставляет *Tomcat* выполнять обратное преобразование DNS для IP-адреса клиента, и это может замедлить все дело. Или вообще остановить. В любом случае, ►►

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ВИРТУАЛИЗАЦИЯ: ЭКОНОМИЯ НА СТОИМОСТИ IT-ИНФРАСТРУКТУРЫ ДО 90%



Единая инфраструктура
на базе свободного
программного обеспечения

Нет лицензионных платежей —
расходы только на внедрение
и техническую поддержку

Минимальные затраты
на оборудование
за счет виртуализации



[www.linuxcenter.ru/shop/
linux-software/office/kitezh](http://www.linuxcenter.ru/shop/linux-software/office/kitezh)

Москва
+7 (499)

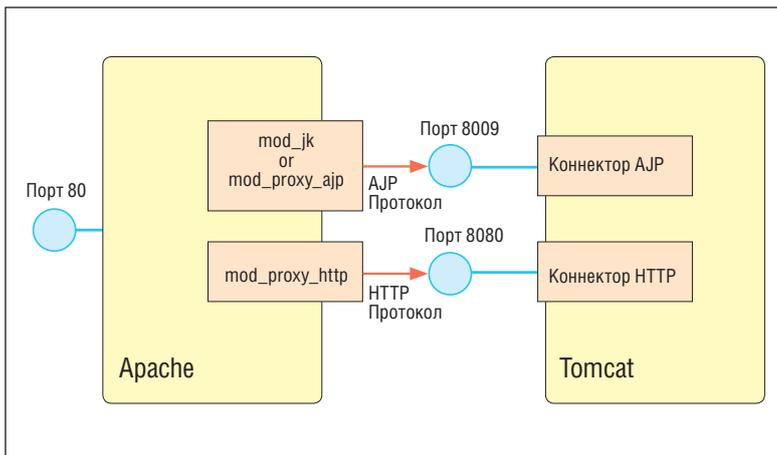
271-49-54

Санкт-Петербург
+7 (812)

309-06-86

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

Linux  center



► **Tomcat часто бывает скрыт за Apache. Серверы могут взаимодействовать друг с другом через AJP или HTTP.**

многие браузеры эффективно скрываются за публичными IP-адресами, предоставляемыми их провайдерами, поэтому доменное имя не скажет вам о клиенте почти ничего.

Также есть вентиль единого входа, который пригодится, если нескольким связанным друг с другом web-приложениям необходима аутентификация, но вы не хотите мучить пользователя, заставляя его заходить в каждое отдельно. Этот вентиль кэширует данные входа (имена пользователей и пароли) на сервере и автоматически подставляет их при доступе к защищенному ресурсу. Он создается так:

```
<Valve className="org.apache.catalina.authenticator.
SingleSignOn" />
```

Наконец, существуют вентили вывода запроса (удобно для отладки) — они выводят заголовки и содержимое куки для всех запросов и ответов, которые могут очень пригодиться для отладки.

Помните, что область действия всех этих вентилях (т.е. применяются ли они ко всему сервису, к одному виртуальному хосту или только к одному приложению) зависит от их расположения в иерархии конфигурационного файла.

Виртуальные хосты

Как я упоминал раньше, для создания дополнительных виртуальных хостов в *Tomcat* достаточно добавить контейнеры `<Host>` в тэг `<Engine>`. Попробуем добавить виртуальный хост для сайта `example.com`. Для начала откройте `/etc/tomcat6/server` и добавьте хост, примерно так:

```
<Host name="example.com"
appBase = "/var/lib/tomcat6/example"
unpackWARs="true"
autoDeploy="true"
<Context path="" docBase="test">
</Context>
</Host>
```

Стоит отметить, что мы задали контейнер `<Context>`, которое помещает содержимое приложения в каталог `"test"` в `appBase`.

Где узнать больше

Если вы установили пакет `tomcat6-docs-webapp`, как я советовал в прошлом месяце, вам обеспечены часы бесконечного веселья на `http://localhost:8080/docs`. А если вы старомодны, как я, и предпочитаете настоящие книги, в которых можно писать на полях, попробуйте «Apache Tomcat 6 для профессионалов» (Одно из руководств для профессионалов от WROX) Чопры [Chopra], Ли [Li] и Дженендера [Genender]. Я бы также посоветовал сайт `www.tomcatexpert.com`.

Теперь создайте новые каталоги:

```
# cd /var/lib/tomcat6
# mkdir -p example/test
```

Чтобы проверить идею на практике, добавим в приложение JSP-страницу. Это файл `/var/lib/tomcat6/example/test/greet.jsp`:

```
<html>
<head>
<title> Welcome to example.com </title>
</head>
<body>
<%
out.println("This is the example.com site");
%>
</body>
</html>
```

В «реальном мире» (все собираюсь съездить туда: говорят, там здорово) нам понадобились бы записи DNS, которые связывают наше доменное имя с IP-адресом нашего сервера. Но пока можно просто добавить запись в `/etc/hosts` таким образом:

```
127.0.0.1 example.com
```

После этого попробуйте попинговать `example.com`, с целью убедиться, что этот адрес преобразуется в `127.0.0.1`:

```
# ping -c 1 example.com
```

Теперь перезапустите *Tomcat*:

```
# service tomcat6 restart
```

Наконец, зайдите на `http://example.com:8080/test/greet.jsp`. Ваши труды будут вознаграждены.

Интеграция Tomcat с Apache

Хотя *Tomcat* прекрасно может работать один, на многих сайтах он прячется за *Apache*. При этом сервер *Apache* выполняет одни запросы сам, а другие перенаправляет на сервер *Tomcat*, который может находиться на том же или на другом компьютере. Этот подход можно использовать, например, для расширения существующего сайта, где используются другие серверные скрипты, такие как CGI или PHP. Он также используется на громоздких сайтах, где за одним клиентом *Apache* скрывается целое семейство серверов *Tomcat*. Для взаимодействия *Apache* и *Tomcat* могут использоваться два протокола. Первый — AJP (Apache JServ Protocol), это двоичный протокол. Второй — старый добрый HTTP. На стороне *Apache* AJP поддерживается модулем `mod_proxy_ajp`, а HTTP — модулем `mod_proxy_http`. Обоим протоколам также необходим модуль `mod_proxy`.

Для связи пространства имен указанной части адреса *Apache* с пространством имен сервера *Tomcat* используется директива `ProxyPass` в файле настройки *Apache*. Для HTTP это может выглядеть так:

```
ProxyPass /stuff/foo/ http://localhost:8080
```

После этого локальный запрос к `http://example.com/stuff/foo/bar` будет преобразован в прокси-запрос `http://localhost:8080/bar` — то есть запрос к серверу *Tomcat*. Чтобы сделать то же самое для протокола AJP, нужна примерно такая директива:

```
ProxyPass /stuff/foo ajp://localhost:8009
```

В обоих случаях *Apache* по сути является обратным прокси. (Я писал о прямом и обратном прокси пару месяцев назад, в `LXF170`.)

Мы вихрем промчались по довольно сложной теме. Надеюсь, я показал достаточно, чтобы вы смогли написать свою программу «Hello World» в среде *Tomcat/Servlet*. Но это не все. Далеко не все. Я не рассказывал о кластерах, о балансировке нагрузки и даже о базовой аутентификации пользователей. Иногда мне кажется, что Linux похож на фрактал Мандельброта — как бы глубоко мы ни заходили, всегда можно зайти еще глубже. Я не знаю, что будет в следующем месяце, но будьте уверены — идей у меня предостаточно! **LXF**

Наши эксперты помогут вам с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Подтвердивший
свою квалификацию
физик.

STIX для всех, но без кириллицы

Я не могу заказать блюдо в ресторане потому, что постоянно смотрю на шрифты в меню.
Дональд Эрвин Кнут

Шрифт STIX (www.stix-fonts.org) — свободный и распространяется под лицензией Open Font License (OFL 1.1). В разработке он уже почти 10 лет, и его спонсоры — американские Химическое, Математическое и Физическое общества (ACS, AMS и APS), Американский институт физики (AIP) и издательство при нем, крупное научное издательство Elsevier и Институт инженеров электротехники и электроники (IEEE).

Зачем им это?

Передача знаний от человека к человеку резко ускорилась, когда Иоганн Гутенберг применил для книгопечатания подвижные литеры. Во второй половине XX века печатные технологии стали полностью цифровыми, и роль литер взяли на себя шрифты. Качество шрифтов — это важно, особенно для научных и технических текстов. Математическая нотация позволяет на порядки сокращать объемы сочинений. При такой степени упаковки качество шрифта по сути определяет и качество передачи информации. STIX готовится как единый набор всех математических и технических символов. Увы, для русских текстов он не годен — символы кириллицы там добавлены чисто для галочки, и им нужна серьезная доработка. Нужен герой, опять, и если он попросит за свое геройство денег, я ему их дам. 50 евро точно дам, а если удавлю жабу, то и больше. Ждем объявления на kickstarter? e.m.baldin@inp.nsk.su

В этом месяце вы научитесь...



Писать на Scratch 66
Бен Эверард представляет визуальный язык программирования, чтобы вовлечь в это дело детишек.



Создавать скрипты 70
Джонатан Робертс научит вас автоматизировать рутину на своей Linux-машине скриптами *Bash*.



Сохранять данные 74
Duplicity умеет сохранять резервные копии ваших папок. **Нейл Ботвик** открывает возможности программы.



Творить музыку 76
Грэм Моррисон поможет вам разобратся в *Ardour* — лучшем приложении для создания музыки в Linux.



Думать о детях 80
Нейл Ботвик разъясняет, как уберечь малышей от столкновения с неприятными сторонами Интернета.



Вникать в Erlang 84
Нет предела совершенству — **Андрей Ушаков** упорно шлифует многозадачные версии функций.



Вычислять параллельно 88
Андрей Сухарев, Андрей Грозин и Евгений Балдин упражняются в построении всевозможных кластеров.



Ретушировать фото 92
Александр Толстой считает фото в коробках из-под ботинок пережитком прошлого века, но выбрасывать их не советует.

Код в учебниках

Строки исходного кода помещаются в цветные плашки. Если строка кода не умещается в колонке, ее остаток переходит на следующую строку, в той же плашке:

```
procedure TfrmTextEditor.  
  mniWordWrapClick  
A плашки разделены зазорами:  
begin  
  mniWordWrap.Checked := false
```

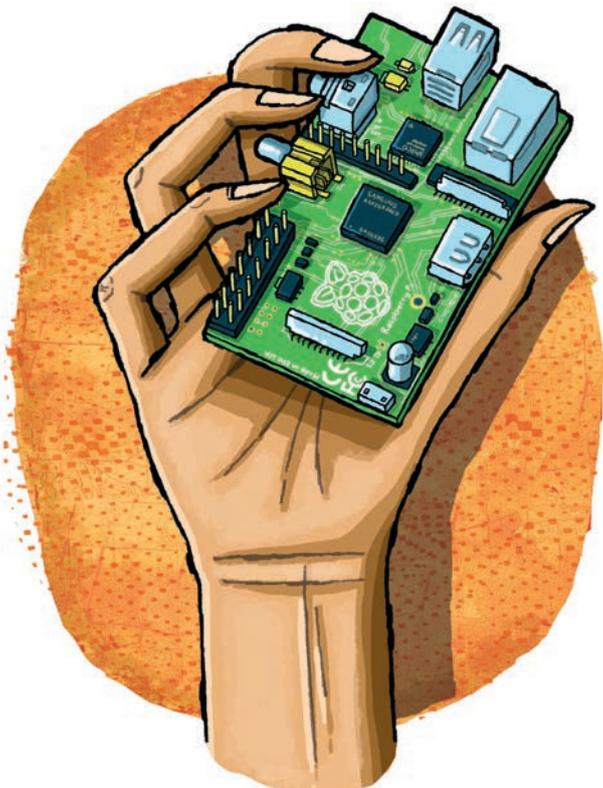
Scratch: Ваяем

Бен Эверард показывает, как написать игру «кошки-мышки» на этом простом языке программирования для RPi.



Наш эксперт

Бен Эверард оставил работу IT-консультанта и на два года отправился в Танзанию устанавливать системы на базе Ubuntu в местных школах. Теперь его знания находят применение в бурлящем котле открытий — Башнях Linux Format.



На Raspberry Pi можно работать с самыми разными языками программирования, но здесь мы воспользуемся Scratch. Это отличный язык для начинающих, так как он иллюстрирует многие идеи программирования, в то же время оставаясь удобным. Он особенно хорош для создания графических программ, например, игр. Если вы никогда не программировали или не программировали графику, мы аккуратно введем вас в курс дела, и, надеюсь, вам понравится. Без дальнейших предисловий, начинаем.

Scratch вы найдете на рабочем столе дистрибутива Raspbian, так что устанавливать ничего не нужно — просто щелкните на иконке, чтобы запустить его. В целях нашего урока лучше пользоваться именно Raspbian, а не другими дистрибутивами для Raspberry Pi, поскольку из других дистрибутивов Scratch поддерживают немногие.

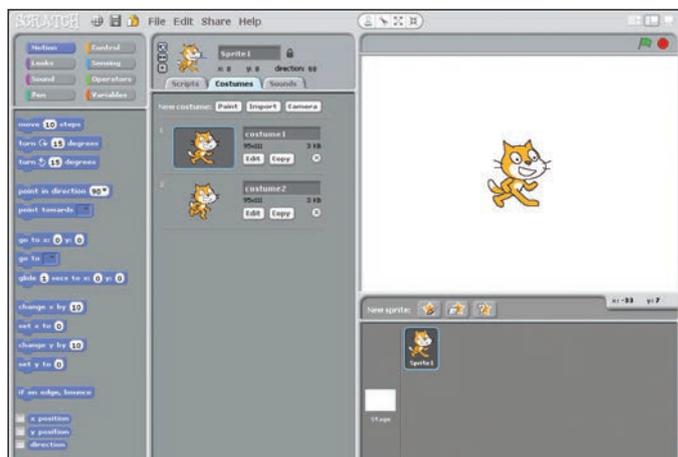
Кто хочет попробовать это на других компьютерах (не с Raspberry Pi) — Scratch доступен в большинстве систем, включая Windows, Mac OS X и основные дистрибутивы Linux. Если вы захотите установить Scratch на другой компьютер, подробности можно узнать на <http://scratch.mit.edu>.

Главное окно разбито на части. В целом, компоненты, из которых собираются программы, находятся слева, программы собираются в середине, а запускаются справа. Каждая программа состоит из спрайтов (изображений), которые содержат скрипты. Эти скрипты управляют тем, что происходит во время выполнения программы.

В левом верхнем углу вы видите восемь слов: Motion [Движение], Looks [Отображение], Sound [Звук], Pen [Ручка], Control [Управление], Sensing [Считывание], Operations [Операции] и Variables [Переменные]. Каждое из них — это категория, содержащая элементы, которые можно перетаскивать в область скриптов, составляя программы. Удачи!



» В Scratch есть набор изображений, пригодных в качестве спрайтов. Для их просмотра выберите New Sprite From File [Новый спрайт из файла] или Costumes > Import [Костюмы > Импорт].



» У каждого спрайта может быть несколько костюмов, или изображений. Чтобы создать новый или управлять имеющимися, перейдите на вкладку Costumes [Костюмы].

Переменные и сообщения

Рано или поздно вам придется заставить свою программу что-нибудь запомнить. Это может быть число, текст или что-то еще. Тут пригодятся переменные. Это небольшие участки памяти компьютера, где ваша программа может помещать данные. На шаге 3 мы создали пару переменных для хранения чисел, хотя в них можно было бы записать и текст. Учтите, что в некоторых языках программирования для хранения различных типов данных нужно создавать переменные различных типов,

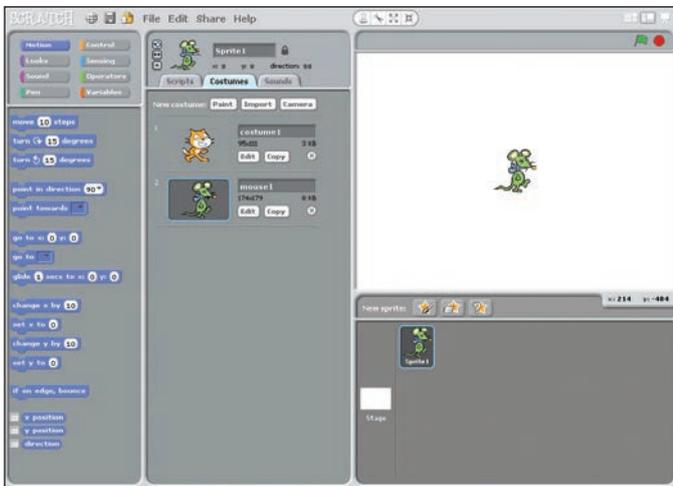
но в Scratch об этом беспокоиться не нужно. После создания переменной ею можно воспользоваться несколькими способами. Во-первых, ей нужно присвоить определенное значение, затем ее значение можно будет проверить в условии (мы это сделали на шагах 6 и 11) или вывести на экран.

Сообщения

При создании нескольких скриптов они могут взаимодействовать друг с другом. Иногда это можно

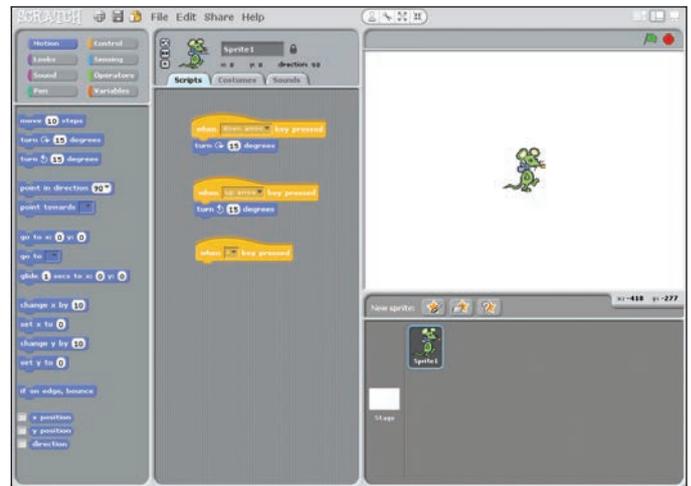
сделать с помощью переменных, но часто лучше бывает использовать сообщения. С их помощью можно запускать скрипты точно так же, как нажатием клавиш. Когда один скрипт отправляет сообщение, оно запустит все скрипты, которые начинаются с When I Receive ... [Когда я получаю...] Как и у переменных, у сообщений есть имена, поэтому сначала их нужно создать, а чтобы скрипт запускался, он должен быть привязан к тому сообщению, которое было отправлено.

Шаг за шагом



1 Создаем мышь

Замените кота на мышь, выбрав Costumes > Import > Animals > Mouse 1 [Костюмы > Импорт > Животные > Мышь 1]. Затем уменьшите размер спрайта, щелкнув на иконке Shrink sprite [Сжать спрайт] (она обведена кружочком) и затем на мыши. Мы сделали ее примерно такого же размера, как миниатюра.



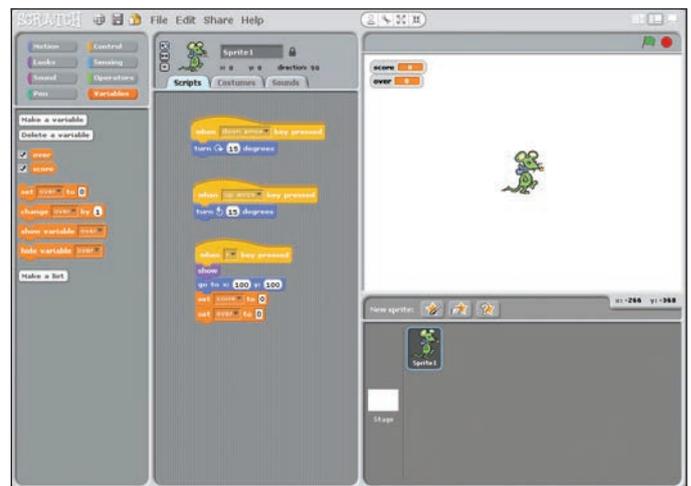
2 Назначаем клавиши

Нажмите на Scripts [Скрипты] и измените When Right Arrow Key Pressed [При нажатии клавиши «стрелка вправо»] на When r Key Pressed [При нажатии клавиши r]. С помощью этой клавиши мы будем начинать новую игру (r [reset] означает «сброс»). Затем переместите Move 10 Steps [Перейти на 10 шагов] в нижнюю часть скрипта. Если перетащить его в левую сторону, блок будет удален.



3 Создаем и называем переменную

Нажмите на Variables [Переменные] в левом верхнем углу (подробно о них можно узнать во врезке). Нажмите на Make A Variable [Создать переменную] и введите имя переменной score. Повторите эти действия и создайте переменную over.



4 Сбрасываем баллы

Под блоком When r Key Presses добавьте блоки show [Показать] (из раздела Looks), Go To X:100, Y:100 (из раздела Motion; не забудьте изменить нули на сотни), Set Score To 0 и Set Over to 0 (оба — из раздела Variables).



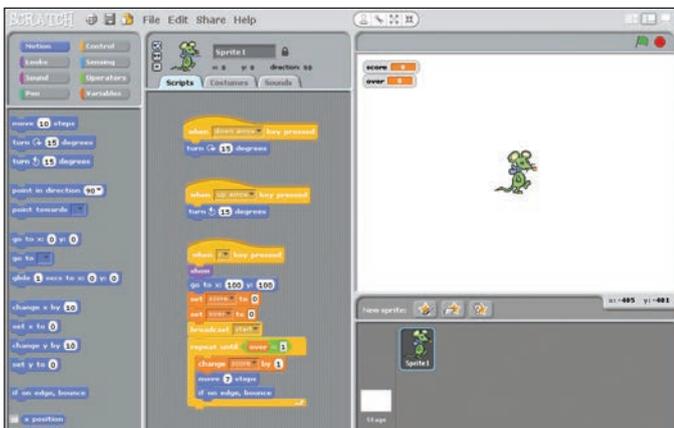
5 Добавляем отправителя

Добавьте блок Broadcast ... [Отправить...] в конце скрипта When r Key Pressed. После этого щелкните на выпадающем меню и выберите New... [Создать], укажите сообщение и назовите его start. Таким образом мы сообщим другому спрайту, что игра начинается.



6 Создаем цикл

Можно создавать циклы, которые будут проходить по одному и тому же коду много раз. Продолжите скрипт блоком Repeat Until ... [Повторять до тех пор, пока...] (из раздела Control), затем перетащите блок ... = ... (из раздела Operators), затем блок Over (из Variables) в левую часть равенства, а в правую введите 1.



7 Добавляем действия в цикл

Внутри блока Repeat Until Over = 1 [Повторять до тех пор, пока не окажется Over = 1] добавьте блоки Change score By 1 [Увеличить score на 1] (из Variables), Move 7 Steps [Переместиться на 7 шагов] (из Motion) и If On Edge, Bounce [Если на краю, отскочить] (также из Motion). Эти три кусочка кода будут повторяться постоянно, пока значение переменной over не станет равным 1.



8 Прячем мышь

Когда игра закончилась (и кот поймал мышь), цикл Repeat Until закончится, и программа продолжит выполняться под ним. Перетащите блок Hide [Спрятать] (из раздела Looks) под цикл, так чтобы мышь исчезла.



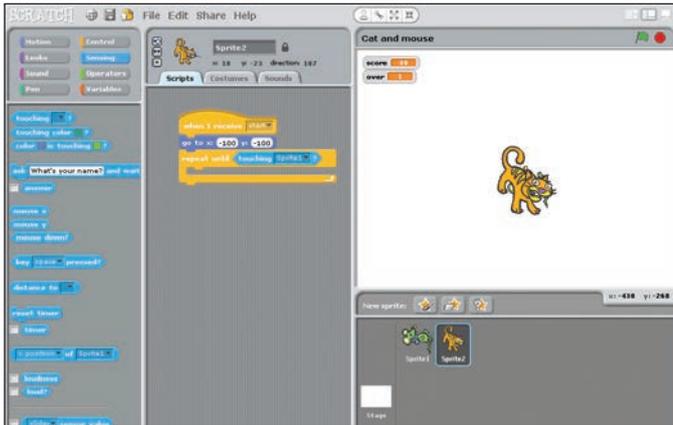
9 Изменяем размер кота

Выберите Choose New Sprite From File > Cat 4 [Выбрать новый спрайт из файла > Кот 4] и уменьшите спрайт до подходящего размера, как мы делали с мышью. У каждого спрайта есть собственный набор скриптов. Переключиться между ними можно, щелкнув на соответствующую иконку в правом нижнем углу.



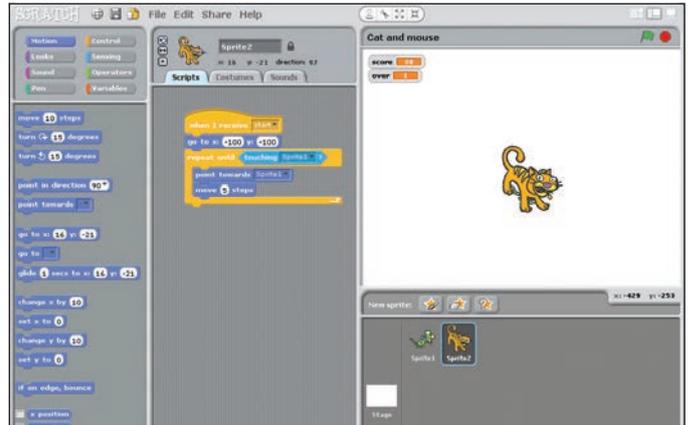
10 Перемещаем кота

В скриптах нового спрайта создайте новый скрипт с командами When I Receive start [Когда я получаю команду на запуск] (из Control) и Go To X:-100 Y:-100 [Перейти к X:-100 Y:-100]. Он переместит кота в противоположный от мыши угол экрана. Середина окна имеет координаты (0,0).



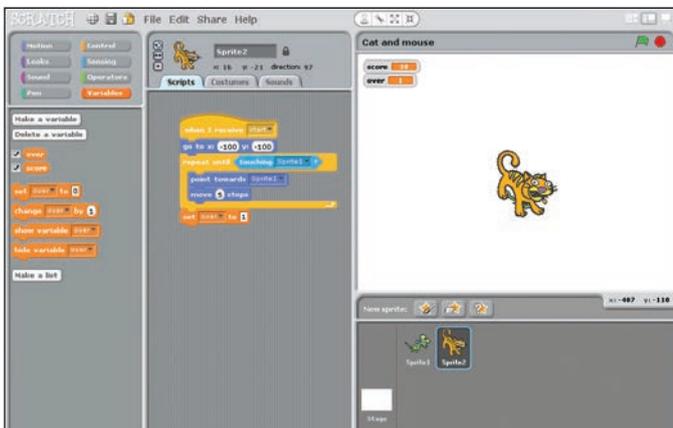
11 Создаем цикл для кота

Как и мыши, коту тоже нужен цикл. Добавьте блок Repeat Until (из раздела Control), а затем в свободном месте добавьте Touching Sprite 1 [Касание спрайта 1] (из Sensing). Цикл будет работать, пока кот (спрайт 2) не поймает мышшь (спрайт 1).



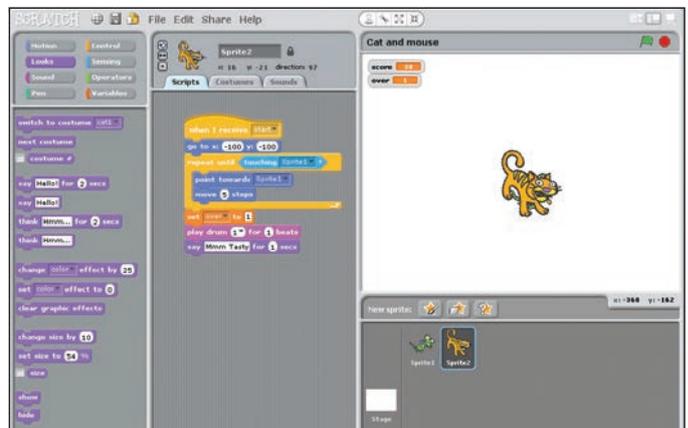
12 Задаем сложность

Внутри блока Repeat Until добавьте Point Towards Sprite 1 [Направить в сторону спрайта 1] (из Motion) и Move 4 Steps [Перейти на 4 шага] (также из Motion). Количество шагов, на которые перемещаются кот и мышшь в каждом цикле, влияет на сложность игры. У нас хорошо работали варианты 4 и 7 соответственно.



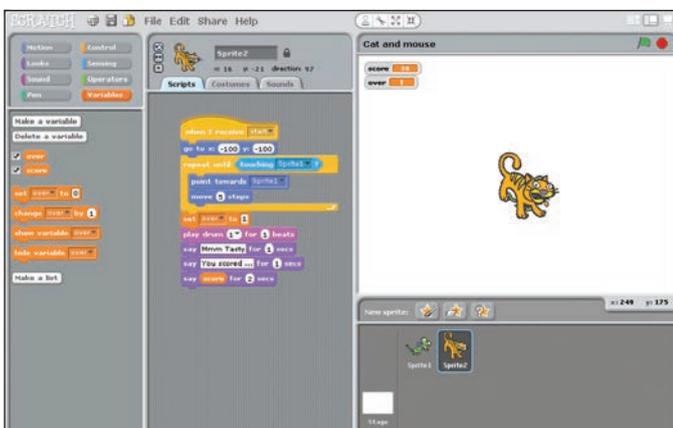
13 Заканчиваем цикл

Цикл закончится, когда кот поймает мышшь — игра закончена, поэтому нам нужно остановить скрипт на спрайте 1. Для этого добавим блок Set over To 1 [Установить over в 1] (из раздела Variables) под блоком Repeat Until. Это остановит главный цикл спрайта 1.



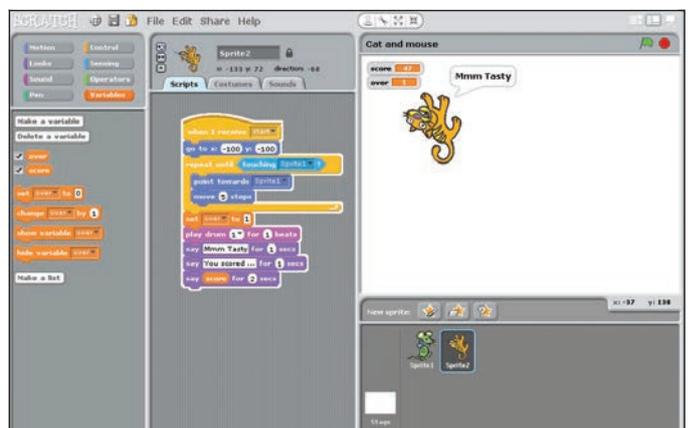
14 Говорим игроку, что игра окончена

Теперь нужно сообщить игроку, что игра закончена. Это мы будем делать в двух видах: звуковым сигналом и сообщением на экране. Добавьте Play Drum 1 for 1 Beats [Сыграть 1 такт Ударных 1] (из раздела Sound), затем Say Mmmm Tasty For 1 Secs [Говорить «М-мм, вкусно» в течение 1 секунды] (из Looks).



15 Отображаем счет

Наконец, мы можем сообщить игроку счет. В каждом цикле мы увеличивали значение переменной score на единицу, поэтому она продолжит расти. Добавьте Say You Scores ... For 1 Secs [Произнести счет... в течение одной секунды], затем добавьте еще один блок Say ... for 1 Secs [Произнести... в течение одной секунды] и перетащите переменную score (из Variables) в свободное поле.



16 Играем!

Нажмите г и играйте! Перемещать мышшь можно стрелками вверх и вниз. Меняя размер спрайтов и величину их перемещения в каждом цикле, можно упрощать или усложнять игру. Удачи и приятной игры! LXF



Резервные копии

Надоело делать резервные копии? **Джонатан Робертс** показывает, как написать несколько коротких и простых скриптов, которые сделают эту работу.



Наш эксперт

Джонатан Робертс сбежал из Башен *Linux Format*, чтобы поискать счастья в качестве сисадмина.



Почти в каждом номере *Linux Format* есть напоминание, как важно регулярно делать резервные копии (а если в этом номере еще не было, то вот, получайте). Но я готов поспорить, что их делают меньше половины из вас. Резервные копии делать нудно, поэтому вы быстро забываете о них, когда ваш мозг переключается на что-то более интересное.



➤ Именно возможность легко сочетать все мощные утилиты GNU делает написание скриптов в *Bash* таким удовольствием.

Есть много других скучных задач, которые регулярно должны выполнять системные администраторы и которыми вы предпочли бы не заниматься — например, удаление старых резервных копий или проверка состояния серверов, установка и настройка новых серверов... список можно продолжить.

Но в Linux все эти рутинные задачи можно решать гораздо проще благодаря мощной оболочке *Bash* и скриптовому языку, которые в большинстве дистрибутивов есть по умолчанию. Поэтому в этой статье мы (еще раз) расскажем вам о *Bash*, сосредоточившись на скриптах, которые позволяют автоматизировать почти любую задачу.

Коротко

Bash — сокращение от “Bourne Again Shell [другая оболочка Борна]”, и впервые она появилась в 1989 году как замена Bourne Shell. Но это мало вам поможет, если не знать, что такое оболочка.

А оболочка — это командный процессор, предоставляющий текстовый интерфейс, с помощью которого вы, пользователь, можете отправлять команды компьютеру. Оболочка интерпретирует эти команды и передает их операционной системе, в которой работает. Любая информация, являющаяся результатом вашей команды, отображается в окне оболочки, чтобы вы интерпретировали ее и, возможно, предприняли какие-то дополнительные действия.

Если вы никогда не пользовались оболочкой (а в этой статье мы будем пользоваться ею совместно с *Bash*), самое время попробовать. На современных компьютерах с Linux оболочка чаще всего запускается как обычное настольное приложение, и в Fedora и Ubuntu оно называется *Terminal*. Найдите и запустите его.

Основы Bash

После запуска появится окно со строкой приглашения в левом верхнем углу. Если курсор в ней мигает, это означает, что оболочка готова к приему команд. Попробуйте ввести команду, набрав **ls** и нажав **Enter**. На экране должен появиться список файлов и каталогов в домашнем каталоге, а затем снова появится строка приглашения.

ls — название команды оболочки, а по сути — программы, которая выводит список файлов в текущем каталоге (по умолчанию, в вашем домашнем). Набрав имя команды и нажав **Enter**, вы велите оболочке выполнить ее.

Работу программы можно изменить, передав ей параметры, часто указываемые как **-x** — то есть дефис, за которым следует буква. Например, наберите **ls -l**, и команда выведет более подробную информацию.

У других команд есть позиционные аргументы, которые указывают, как должна работать программа; часто они обязательны. Например, команда **cp**, которая копирует файлы, принимает два позиционных аргумента: имя копируемого файла и имя и местоположение копии:

```
cp hello.txt world.txt
```

Эта команда скопирует файл **hello.txt** в текущем каталоге (по умолчанию, ваш домашний каталог) в файл **world.txt**. Если

— СКРИПТОМ Bash

Найдите свое место

Когда вы пользуетесь оболочкой, она всегда отслеживает текущий каталог, где вы в данный момент находитесь. Как упоминалось в основном тексте, при первом запуске оболочки она по умолчанию делает текущим каталогом домашний.

Эта на первый взгляд незначительная деталь на самом деле очень важна. Новичкам: если вы не указали абсолютный путь, то все пути к файлам и каталогам интерпретируются как относительные по отношению к текущему рабочему каталогу. Если запустить команду из неверного каталога, можно получить неожиданные результаты, а то и удалить важный файл.

Другая причина, по которой это важно, в том, что некоторые операции, такие как размонтирование каталога, завершатся неудачно, если в этом каталоге есть активная оболочка. Это не проблема, если вы работаете в оболочке, выполняя команды, но если вы запустили скрипт, который переходит в каталог, который позже вы пытаетесь размонтировать, скрипт может выдать ошибку — и вы не будете понимать, почему.

Текущий каталог всегда можно определить, запустив команду **pwd** — сокращение от “present working directory [текущий рабочий каталог]”.

файла **world.txt** не существует, он будет создан, а если он есть, то будет перезаписан, так что с этим надо поаккуратнее!

Если надо перейти в другой каталог, можно воспользоваться командой **cd**. Например, команда **cd /** переместит вас в корневой каталог, который находится на самом верху иерархии файловой системы, а **cd /usr/bin** — в подкаталог **bin** каталога **user**, который, в свою очередь, является подкаталогом корневого каталога.

Пути могут быть и относительными: если вы в своем домашнем каталоге и хотите перейти в свой подкаталог **Documents**, наберите **cd Documents** — оболочка найдет каталог **Documents** в текущем каталоге и переместит вас туда.

Наконец, с помощью путей также указываются файлы, необходимые для работы команд. Например, чтобы скопировать файл **hello.txt** в **world.txt**, но файл **world.txt** находится в каталоге **Documents**, а **hello.txt** — в каталоге **test**, можно скомандовать

```
cp test/hello.txt /home/jon/Documents/world.txt
```

Вы также увидите, что можно комбинировать и сопоставлять относительные и абсолютные пути.

Привет, мир

Ну вот, с основами разобрались, и пора перейти к теме нашего урока: написанию скриптов *Bash*.

Рассказывая вам о возможностях оболочки, позволяющих использовать ее в качестве среды программирования, мы упомянем еще много других команд, которые, возможно, будут вам

незнакомы. Чтобы не задерживаться на рассказах о каждой, мы советуем вам пользоваться командой **man** для получения подробной информации о каждой команде. Просто наберите **man <имя команды>**, и вы получите указания по использованию этой команды. Попробуйте набрать **man man**, чтобы узнать, как работает сама команда **man**.

«Этот простой подход уже дает огромные возможности.»

В соответствии с давней традицией руководств по программированию начнем с программы, которая отображает на экране слова “Hello World”. Наберите следующий текст в файл и сохраните его:

```
#!/bin/bash
echo "Hello World"
```

После этого сделайте файл исполняемым:

```
chmod +x <filename>
```

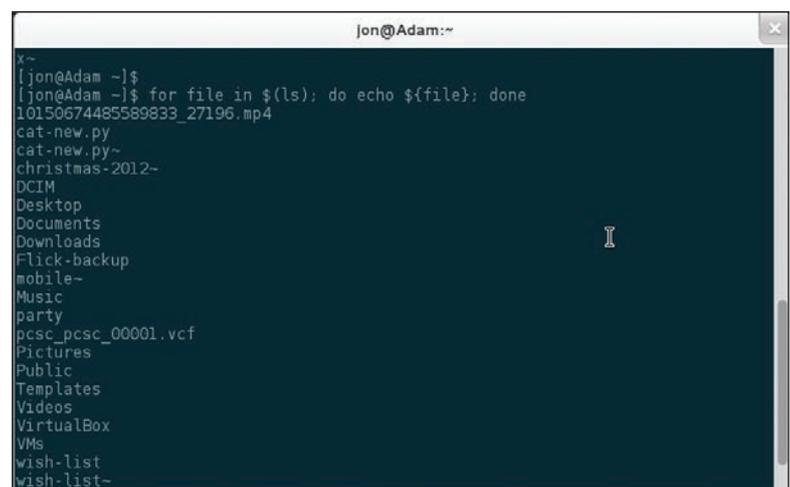
Затем запустите его из оболочки, набрав

```
./<filename>.
```

— и вы должны увидеть эти вошедшие в легенду слова в вашей оболочке.

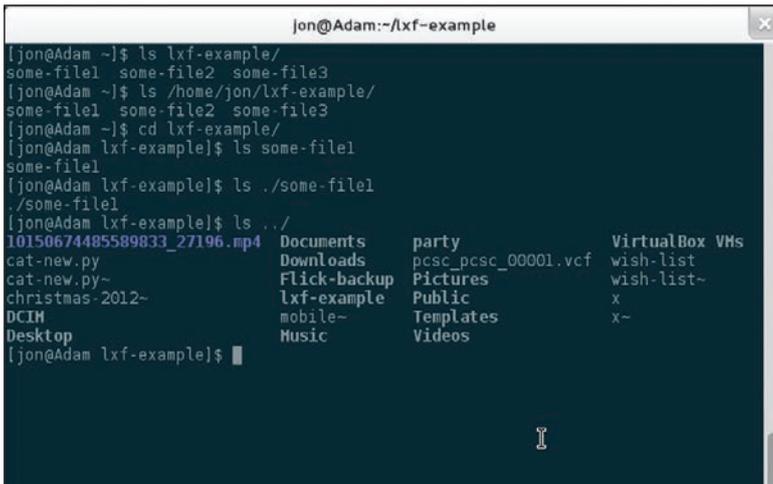
Первая строка — так называемая «шапка». По ней операционная система определяет, с помощью какой программы нужно исполнять содержимое скрипта. В данном случае мы указали полный путь до программы *Bash*. Указываемый здесь путь должен быть абсолютным и иметь символы **#!** в начале.

Вторая строка — стандартная команда оболочки. В данном случае это команда **echo**, которая выводит на экран все



» Команды скриптов *Bash* можно не только записывать в файлы, но и писать прямо в консоли.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



Ссылаться на файлы в Bash можно по-разному, в том числе по абсолютным и относительным путям.

переданные ей аргументы. Здесь можно было поместить любую приглянувшуюся нам команду, и при запуске скрипта оболочка выполнила бы ее.

Это и придает скриптам огромную мощь. Можно воспользоваться всеми командами Bash, а также такими фантастическими функциями, как перенаправление и каналы, которые свяжут команды друг с другом; в скриптах есть и другие способы соединения команд — операторы условия, циклы **for** и функции.

Два последних пункта, которые следует отметить: во-первых, файл нужно сделать исполняемым, иначе операционная система

не позволит запустить его. Во-вторых, имя файла нельзя просто указать, как имя обычной команды: перед ним нужно набрать **./**. Точка — обозначение текущего каталога, а слэш — обычный разделитель файлов и каталогов; собственно говоря, слэш и точка — это абсолютный путь до файла в текущем каталоге (чтобы понять, почему его нужно указывать, загляните во врезку **\$PATH**.)

Переменные

Предыдущий скрипт можно разными способами усложнить. Например, можно добавить новые команды, поместив каждую на новой строке:

```
#!/bin/bash
echo "hello world"
echo "Вот файлы из моей текущей директории:"
ls
```

Со всеми командами, имеющимися в Bash, этот очень простой подход к программированию уже дает вам огромные возможности. Но чтобы сделать что-нибудь по-настоящему мощное и добавить в свои скрипты интеллект и гибкость, нам понадобятся некоторые конструкции, имеющиеся в других языках программирования. Начнем с переменных, с помощью которых можно связать имена и значения. Рассмотрим пример:

```
#!/bin/bash
name=Jon
echo "Hello ${name}"
echo "${name}, какое милое имя."
```

В первой строке скрипта (после «шапки») текстовая строка **Jon** присваивается переменной **name** [имя]. После присваивания можно обращаться к этой переменной в любом месте скрипта,

помещая имя переменной в конструкцию **\${}**. При выполнении скрипта Bash разберет строки и заменит значением **name** каждую конструкцию **\${name}**.

Это и само по себе довольно удобно, но в Bash есть еще несколько гораздо более полезных трюков с переменными. Один из самых полезных — присвоение переменной вывода команды, для этого команду нужно заключить в обратные апострофы:

```
date=`date +%Y-%m-%d`
```

Она назначит вывод команды **date** (2013-07-06 или тот день, в который вы это читаете) переменной **date** для последующего использования в вашем скрипте. Можно поступить еще умнее и добавить вывод команд прямо в другие строки конструкцией **\$()**:

```
echo "Сегодняшнее число $(date +%Y-%m-%d)"
```

Этот пример очень удобен при создании резервных копий, так как позволяет создавать файлы с текущей датой в именах.

Следим за порядком

Еще один пример средств объединения команд Bash, который обычно есть в других языках программирования — средства для управления порядком выполнения команд. До этого момента Bash выполнял команды именно в том порядке, в каком мы их записывали; ну, а если нужно изменить поведение программы в зависимости от некоторых условий? Это позволят сделать условные операторы.

Первый из них — оператор **if...then**. Он проверяет, верно ли некоторое условие, например, существует ли указанный файл, и только если условие верно, выполняет следующий код:

```
#!/bin/bash
if [ 2 > 1 ]; then
echo "True"
fi
```

\$PATH

В выполнении команд в Bash нет ничего таинственного. Когда вы вводите имя команды, оболочка находит исполняемый файл или скрипт, на который ссылается имя, и затем выполняет его, как если бы это был обычный скрипт или программа, выполняемые пользователем, как в **./hello.bash**. Но если не указать абсолютный путь к программе, как оболочка узнает, где искать файл?

Ответ — переменная **\$PATH**. В вашей оболочке есть масса переменных «окружения», которые влияют на то, как в текущей оболочке выполняются программы и интерпретируются команды. Одна из таких переменных называется **PATH** и содержит список каталогов, где могут быть найдены двоичные файлы команд или скрипты.

Каждый раз, когда вы вводите имя команды, оболочка просматривает список каталогов в этой переменной слева направо, пока не найдет файл с тем же именем, что и имя команды, которую вы только что запустили. Если оболочка не найдет файла, она выдаст сообщение об ошибке:

```
-bash: foo: command not found
```

Содержимое переменной **\$PATH** можно просмотреть, как и содержимое любой другой переменной, выведя его на экран: **echo \$PATH**. И так же, как и значение любой другой переменной, его можно изменить. Если вы хотите, чтобы изменения сохранились после завершения текущего сеанса оболочки, нужно добавить их в один из файлов настройки Bash, **.bashrc** или **.bash_profile**, в зависимости от того, относятся ли эти изменения к оболочкам входа в систему, подоболочкам или и к тем, и к другим.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

Это пример тривиальный, так как он всегда будет выводить **True**, но он демонстрирует синтаксис. В *Bash* операторы **if** начинаются с ключевого слова **if**, за которым в квадратных скобках следует проверяемое условие. За ним следует ключевое слово **then**, а далее — команды, выполняемые в том случае, если условие верно; после этого оператор замыкается ключевым словом **fi**.

В *Bash* есть целый набор встроенных проверок, многие из которых очень полезны, например, **-f**, которая проверяет, что файл существует, и **-nt**, которая проверяет, что один файл новее другого:

```
if [ -f /var/run/backup.lock ]; then
exit 1
else
touch /var/run/backup.lock
tar -czf /mnt/backup-$(date +%Y-%m-%d).tgz /home/jon
rm /var/run/backup.lock
fi
```

В этом примере мы проверяем, существует ли файл блокировки. Если да, мы не начинаем новой операции резервного копирования, так как предыдущая еще выполняется. Если файла блокировки нет, мы создаем его в блоке **else**, выполняем резервное копирование и затем удаляем файл блокировки.

Советуем потратить немного времени и взглянуть на ман-страницу *Bash*, чтобы узнать, какие проверки еще доступны.

Хотя мы говорили, что не будем отвлекаться на рассказы об отдельных командах *Bash*, которые встретятся в этой статье, команде **exit** стоит уделить немного больше внимания. Когда программа или команда *Bash* завершается, она возвращает числовое значение. 0 принято считать обозначением успешного завершения команды, а все остальные значения показывают: что-то пошло не так. Такое особенно полезно в сочетании со встроенной переменной **?**, которая хранит код завершения предыдущей команды — это позволяет управлять ходом программы в зависимости от успешного или неудачного завершения ее частей.

Циклы for

Еще одна очень полезная конструкция, которая есть почти во всех языках программирования — цикл **for**. Она используется для итерации по группам объектов, позволяя вам применить один и тот же блок кода к каждому объекту по очереди. В *Bash* она работает подобно выражению **if**:

```
for file in $(ls); do
echo ${file}
done
```

Еще один глупый пример, но он иллюстрирует синтаксис. После ключевого слова **for** идет переменная цикла, которая по очереди будет принимать каждое из значений выражения, следующего за ключевым словом **in**. Ключевые слова **do** и **done** обозначают начало и конец блока, а к переменной **file**, как и к любой другой переменной, можно обращаться внутри цикла.

Последняя возможность, которая есть во всех хороших языках программирования и которая поможет сделать ваши скрипты максимально удобными — функции. В *Bash* создание и вызов функций довольно стандартны. Для определения функции просто напишите ее имя в парной круглых скобок, а затем укажите код функции в фигурных скобках:

```
func () {
echo "Моя первая функция."
echo $1
}
func "hello"
```

Как вы видите по этому примеру, вызов функции происходит точно так же, как и вызов обычной команды *Bash* — вы указываете имя функции и все аргументы, которые хотите ей передать. Обращаться к аргументам внутри функции можно по номерам: **\$1** — первый аргумент, **\$2** — второй, и т.д.

Перенаправления и каналы

Во всех приведенных примерах данные выводились на экран в сеансе оболочки, который в данный момент выполнялся в терминале. Но если вы думаете, что так делают все программы, то вы ошибаетесь.

На самом деле программы направляют свой вывод в «канал» **stdout**. У канала два конца: один из них подключен к программе, генерирующей выходные данные, а другой — к чему-то еще, что принимает и обрабатывает вывод. По умолчанию, это «что-то еще» — ваш экран, но это не обязательно.

Управлять каналами можно двумя способами. Первый — переключить выход канала, чтобы он направлялся в файл, а не на экран:

```
echo "Hello world" > hello.txt
```

Чтобы вам все было понятно, **>** — это символ, который как раз и осуществляет перенаправление.

Есть еще другой канал — **stdin**. Как легко догадаться, он предназначен для управления входными данными программы. Одна из самых мощных возможностей *Bash* — перенаправление потока **stdout** одной программы на **stdin** другой:

```
ls | grep *.txt
```

Здесь символ **|** (чтобы набрать его на британских клавиатурах, нажмите Shift+**|**) направляет вывод команды **ls** на вход команды **grep**, в данном случае отфильтровывая все результаты, заканчивающиеся на **.txt**. Так можно объединить несколько программ, проводя сложные операции над данными.

Точно так же можно обращаться к аргументам, переданным скрипту в командной строке; так вы сможете писать программы, которые работают с конкретными файлами или изменяют свое поведение в зависимости от пользовательских настроек. Представьте себе следующий файл с именем **name.bash**:

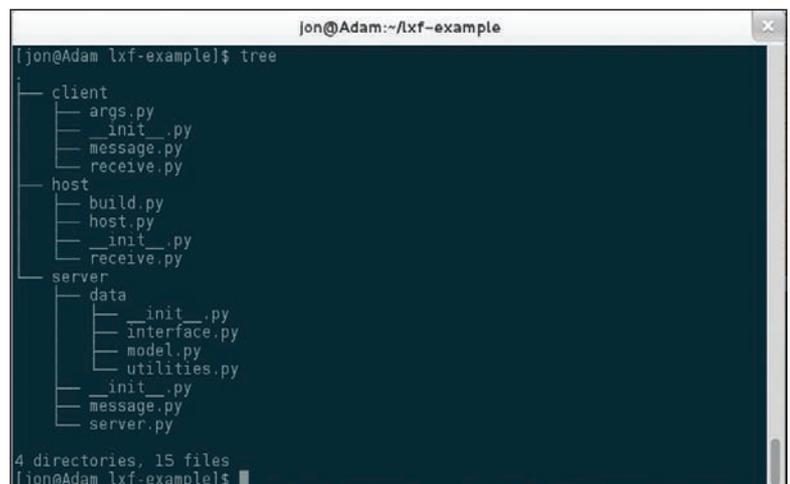
```
#!/bin/bash
echo $1

jon@adam ~$ ./name Jon
Jon
```

Превосходство Борна

Это почти все, на что нам отведено место в этом месяце, но с этими основами у вас есть все необходимое, чтобы писать скрипты для автоматизации любых задач на своем компьютере. Помните, что если вы столкнулись со сложной задачей и не знаете, как ее решить, сперва поищите готовую программу, которая может это сделать, а если не получится, подумайте о том, как объединить несколько простых программ для решения более сложной задачи.

В частности, особенно полезны команды **grep**, **sort** и **cut**, а также **sed** (хотя только о ней можно написать отдельную статью или целую книгу!). **LXF**



► Разбивайте код на фрагменты и храните каждый в собственном файле и каталоге. Это внесет порядок в ваши проекты.

Duplicity: Будем

Продолжая раскапывать «золотую жилу» резервного копирования, **Нейл Ботвик** учит создавать копию каталогов с помощью простой программы.



**Нейл
эксперт**

У **Нейла Ботвика** по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет вам, где находится центральный сервер.

На предшествующем уроке Джонатан показал вам, как написать простой скрипт резервного копирования в *Bash*, но есть и ряд готовых программ, способных на такое же, и у каждой свой подход, свои сильные и слабые стороны. На данном уроке мы познакомимся с *Duplicity* (<http://www.nongnu.org/duplicity>). Не то что другие программы хуже, но *Duplicity* проста, эффективна и поддерживает локальные, сетевые и облачные резервные копии. Начнем с простейшего из примеров: создания резервной копии одного дерева каталогов в локально смонтированной файловой системе, например, на внешнем жестком диске:

```
duplicity ~/Documents file:///mnt/mybackup/Documents
```

Команда **duplicity** принимает два аргумента: место назначения и источник. Один из них — стандартный путь в файловой системе, другой — URL-адрес. URL ссылается на местоположение резервной копии, и если он следует за путем, то *Duplicity* знает, что нужно выполнить резервное копирование. Если указать сначала URL, а затем путь, *Duplicity* восстановит содержимое локального каталога из указанной резервной копии. Чтобы задать в качестве места назначения локальный каталог, используется формат `file://URL`; обратите внимание, что два слэша здесь обозначают сам URL, и если вы захотите указать абсолютный путь, понадобится третий слэш.

У вас спросят пароль, так как по умолчанию *Duplicity* шифрует резервные копии. Не забывайте его: без него вам не восстановить файлы. Запустив эту команду и затем заглянув в каталог резервной копии, вы увидите несколько файлов, начинающихся с `duplicity-full.20130319T133101Z.vol1.diff.tar.gpg`. Такое название означает, что это полная резервная копия и процесс копирования был запущен в указанные день и время. Резервная копия

```
[nelz@hactar ~]$ cat /home/.duplicityrc
/home/nelz/.claws-mail/imapcache
/home/nelz/Archive
/home/nelz/Documents
/home/nelz/.claws-mail/tagsdb
/home/nelz/.claws-mail/RSSyl
/home/nelz/.mozilla/firefox/hawfl2tb.default/Cache/*
/home/nelz/.cache/duplicity
/home/nelz/.cache
/home/.zfs
**/*.iso
**/unison.log
**/t
**/.thumbnails
**/.kde4/share/config/session
**/.kde4/cache-*
**/.kde4/socket-*
**/.kde4/tmp-*
/home/nelz/.claws-mail/RSSyl
**/.googleearth/Cache
**/.mythtv/cache
/home/nelz/.wine
**/.mozilla/firefox/default.*/Cache
**/.histfile.*
**/.local/share/Trash
```

» Списки включения и исключения позволяют создать копию только нужных файлов, исключив множество временных файлов.

разбивается на файлы по 25 МБ, что можно изменить параметром `--volsize`; если резервная копия будет находиться в локальной файловой системе, стоит увеличить этот размер, чтобы файлов было меньше. Файлы меньшего размера удобны при копировании на удаленный компьютер.

Инкрементные копии

Итак, мы сделали резервную копию важного каталога или даже всего домашнего каталога, но она занимает почти столько же места, сколько оригинал. При шифровании *GPG* сжимает данные, но самые большие файлы — видео, фотографии и т. д. —

Пропускаем мусор

Пока мы делали резервные копии целых каталогов, но в них могут быть файлы, которые копировать незачем. Объем моего каталога `.cache` — около 750 МБ, и большую их часть съел *Chromium*. Эти данные не нужно копировать, так что их можно исключить — командой

```
duplicity --exclude ~/.cache src dest
```

Параметр `--exclude` можно указывать несколько раз, но обычно проще поместить список исключений в файл и указать его *Duplicity*.

```
duplicity --exclude-globbing-filelist ~/.duplicityexcludes
```

Указанный файл содержит следующее:

```
- /root/.cache
- **/*.tmp
```

Маски — это стандартные маски оболочки, а поставив в начале строки `+`, можно принудительно включить ее в копию, даже если в последующей строке она будет исключена:

```
+ **/.cache/duplicity
- **/.cache
```

Эти две строки исключают из копирования каталог `.cache`, кроме подкаталога *Duplicity*. Пути должны начинаться с `**`, что делает их относительными по отношению к источнику, указанному в командной строке, или абсолютными. Еще один параметр для исключения — `--exclude-other-filesystems`, его можно использовать для копирования резервного каталога и исключения `/dev`, `/proc` и `/sys`, а также всего, смонтированного в `/mnt`. Включение в список копирования вашего внешнего диска, на который и делается копия — рецепт катастрофы.

Копии через сеть

Во всех предыдущих примерах мы копировали данные в локальный каталог, но *Duplicity* может делать это и по сети. Чтобы воспользоваться SSH, запустите команду:

```
duplicity --ssh-backend=pexpect --sshoptions="
-oldidentityfile=~/.ssh/id_ecdsa.pub" /src/dir
ssh://user@hostname//dest/dir
```

С клиентом SSH по умолчанию у меня были проблемы, и параметр `pexpect` решил их; вы можете попробовать запустить его с этим параметром и без и посмотреть, какой вариант работает лучше. Параметр `IdentityFile` — это публичный ключ SSH, который также нужно скопировать в файл `authorized_keys` на удаленном компьютере, если вы не хотите каждый раз вводить пароль от SSH. Слэш после имени хоста отделяет его от следующего за ним пути, который является относительным по отношению к домашнему каталогу пользователя на удаленном компьютере, так что для абсолютного пути нужно указать еще один слэш.

Для подключения по SSH *Duplicity* использует SFTP, поэтому убедитесь, что в вашем сервере SSH он включен (обычно это так), в противном случае придется включать `scp` параметром `--use-scp`, что чуть менее эффективно. *Duplicity* также может работать с Amazon S3, Google Docs and FTP и другими сервисами. Подробности см. на `man`-странице.

КОПИРОВАТЬ

обычно уже сжаты, поэтому особой экономии места не получают. А если через неделю вы захотите сделать еще одну резервную копию? Если делать такие резервные копии регулярно, у вас либо кончится место, либо придется делать копии реже, либо придется удалить все резервные копии, кроме самых последних — все это плохие решения. *Duplicity*, как и многие другие программы для резервного копирования, помогает решить эту проблему, разделяя резервные копии на полные и инкрементальные. Полная резервная копия, как ясно из термина, содержит все файлы, тогда как инкрементальная — только файлы, которые изменились с момента создания предыдущей резервной копии. Это означает, что можно делать полную копию, скажем, раз в месяц, а инкрементальные — каждый день, и при этом занимать не слишком много места на диске. Важно и то, что можно восстановить эти файлы по состоянию на любой день (более подробно о восстановлении позже).

Тип создаваемой резервной копии можно указать перед путями таким образом:

```
duplicity full ~/Documents file:///mnt/mybackup/Documents
```

или

```
duplicity incremental ~/Documents file:///mnt/mybackup/Documents
```

Возможно, вы удивляетесь, почему мы не сделали этого раньше. Причина в том, что *Duplicity* достаточно умна, чтобы понять, что делать. Если в каталоге назначения уже есть полная резервная копия, она выполняет инкрементальную копию, в противном случае — полную. Если копирование выполняется через *Cron*, есть параметр `--full-if-older-than`, который принудительно делает полную резервную копию, если существующая полная копия старше указанного времени. Если данные в каталоге изменяются часто, то в долгосрочных инкрементальных копиях может быть много ненужных данных, поэтому можно воспользоваться командой:

```
duplicity --full-if-older-than 2W ~/Documents file:///mnt/mybackup/Documents
```

чтобы создавать полную копию каждые две недели. Для указания периода времени используется число с обозначениями D (дни), W (недели) и M (месяцы).

Восстановление и чистка

Мы рассмотрели различные параметры для создания резервных копий. Если все хорошо, то оно и достаточно, но если что-то пойдет не так, нужно знать, как восстановить данные. Для восстановления из резервной копии достаточно поменять местами аргументы источника и места назначения — если первый аргумент начинается с обозначения URL: `file://`, `ssh://` и т. д., то *Duplicity* поймет, что вы хотите восстановить данные. Для восстановления конкретного файла или каталога скомадите

```
duplicity --file-to-restore path/to/file backup_url restore_dir
```

Если хотите восстановить более старую версию файла, воспользуйтесь параметром `-time` — например, для восстановления файла в версию четырехдневной давности подойдет команда

```
duplicity --time 4D --file-to-restore path/to/file backup_url restore_dir
```

Когда-нибудь у вас израсходуется место (или за него придется платить слишком много), и вы захотите удалить старые резервные копии. Команда

```
duplicity remove-older-than 3M --force backup_url
```

удалит резервные копии давности более трех месяцев, а без параметра `--force` просто выведет их список. Она не удалит резервные

Еще о шифровании

При каждом запуске *Duplicity* запрашивает парольную фразу *GPG*. Этого можно избежать, указав параметр `--no-encryption`, но это плохая идея, если резервная копия хранится на облаке. *Duplicity* может воспользоваться переменной окружения `PASSPHRASE`, если она установлена, но оставлять ее установленной постоянно небезопасно, поэтому делать это можно в командной строке:

```
PASSPHRASE="asdf" duplicity --options source destination
```

Пока *GPG* применялась для симметричного

шифрования, и для шифрования и дешифрования использовалась одна и та же парольная фраза. Чтобы зашифровать данные своим приватным ключом, нужно указать параметр `--encrypt-key`. При этом в переменной `PASSPHRASE` должна быть парольная фраза ключа. Если вы выберете этот вариант, само собой разумеется, копия ключей *GPG* должна храниться в безопасном месте, отдельно и от резервной копии, и от диска, на который делается резервная копия, иначе вы не сможете восстановить файлы.

копии независимо от их «возраста», если от них зависят более новые версии. Если последняя полная копия делалась полгода назад, *Duplicity* удалит только копии старше этой, так как инкрементальные копии без родительской полной копии бесполезны.

Duplicity проста в использовании, но у нее есть несколько параметров, которые не хочется вводить каждый раз, поэтому стоит создать короткий скрипт для вызова *Duplicity* с необходимыми параметрами, например:

```
#!/bin/sh
export PASSPHRASE="asdf"
duplicity --ssh-backend=pexpect --ssh-options="--oldidentityFile=/root/id_ecdsa.pub" --full-if-older-than 1M --exclude-globbing-filelist /home/.duplicity-excludes /home ssh://my.server/duplicity/home
duplicity remove-older-than 3M --force --ssh-backend=pexpect --ssh-options="--oldidentityFile=/root/id_ecdsa.pub" ssh://my.server/duplicity/home
```

Далее этот скрипт можно вызывать из *Cron*, чтобы процесс был полностью автоматизирован. **LXF**

> Duplicity обычно выводит довольно мало данных; чтобы их было больше, воспользуйтесь ключом -v.

```

import of duplicity.backends.hsibackend Succeeded
import of duplicity.backends.gdcbkbackend Succeeded
import of duplicity.backends.localbackend Succeeded
import of duplicity.backends.ftpbkbackend Succeeded
import of duplicity.backends.imapbkbackend Succeeded
import of duplicity.backends.rsynckbackend Succeeded
import of duplicity.backends.sshbkbackend Succeeded
import of duplicity.backends.ulbkbackend Succeeded
Main action: inc

=====
duplicity 0.6.21 (January 23, 2013)
Args: /usr/bin/duplicity-python2.7 -v8 --archive-dir=/var/cache/duplicity --ssh-backend=pexpect --exclude-other-filesystems --volsize=50 --ssh-options="--oldidentityFile=/root/.ssh/id_ecdsa" --name usr --full-if-older-than 7D /usr sftp://root@lunkwill:/foo/duplicity/hactar/usr
linux hactar 3.8.6-gentoo #1 SMP Sat Apr 6 14:31:00 BST 2013 x86_64 Intel(R) Core(TM) i7-2600K CPU @ 3.40 GHz
/usr/bin/python2.7 2.7.3 (default, Jan 11 2013, 09:39:34)
[GCC 4.6.3]

=====
Using temporary directory /tmp/duplicity-x8FVjY-tempdir
Temp has 8396832768 available, backup will use approx 68157440.
Running 'sftp --oldidentityFile=/root/.ssh/id_ecdsa' -oServerAliveInterval=15 -oServerAliveCountMax=2 root@lunkwill' (attempt #1)
sftp command: 'mkdir "/foo"'
sftp command: 'cd "/foo"'
sftp command: 'mkdir "duplicity"'
sftp command: 'cd "duplicity"'
sftp command: 'mkdir "hactar"'
sftp command: 'cd "hactar"'
sftp command: 'mkdir "usr"'
sftp command: 'cd "usr"'
sftp command: 'ls -l'
Local and Remote metadata are synchronized, no sync needed.
Running 'sftp --oldidentityFile=/root/.ssh/id_ecdsa' -oServerAliveInterval=15 -oServerAliveCountMax=2 root@lunkwill' (attempt #1)
sftp command: 'mkdir "/foo"'

```

Ardour 3: Как

Личный Джастин Бибер команды LXF — Грэм Моррисон погружается в удивительный мир создания звука.



Наш эксперт

Когда Грэм Моррисон не ломает голову над тем, что поставить в Linux Format, и не возится со старыми синтезаторами, он скорее всего ищет способы повысить продуктивность KDE 4.



Мы долго ждали *Ardour 3*, невероятно мощную программу для обработки звука. Путь ее развития мы описали в прошлом номере, но приучить *Ardour* не так-то просто. Как и среда трехмерного моделирования *Blender*, *Ardour* предлагает лучшие в своем классе возможности, но не идет ни на какие компромиссы в удобстве работы. Это значит, что новичкам

начать работу с ней будет непросто. На следующих страницах мы предпримем все усилия, чтобы справиться с этим — расскажем начинающим об основах, чтобы каждый мог записывать, микшировать и выполнять мастеринг своих аудиозаписей.

Готовим «железо»

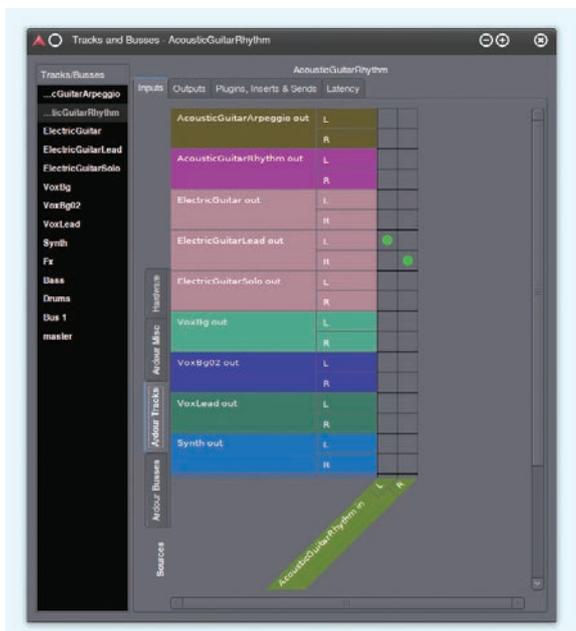
Ключ к успеху при работе со звуком в Linux — выбор лучшего доступного оборудования. Вечная борьба за стабильность и качественный звук или простое подключение устройства и запись — это две большие разницы. С приложением профессионального уровня типа *Ardour* стоит использовать внешнюю звуковую карту, а не встроенную в материнскую плату. Это не только скажется на качестве звука, поскольку на карту не будет влиять работа прочих устройств компьютера, но и карта в целом будет работать хорошо и безотказно. Но вы вполне можете воспользоваться и встроенной звуковой картой: так вы попробуете программу и почувствуете ее возможности. Ограничив себя только цифровыми источниками звука, например, программным синтезатором на рабочем столе, вы сможете создать звук профессионального качества, но высококачественных записей у вас не получится.

Кроме встроенных устройств, существуют еще три вида звуковых интерфейсов. Первый использует PCI-слот на материнской плате, и этот вариант не слишком удобен для записи в нестационарных условиях, для ноутбуков и для небольших компьютеров. Исторически это такие карты, как Creative SoundBlaster, а также более профессиональные модели вроде RME. Вам нужно убедиться, что выбранное устройство поддерживает Linux — для карт SoundBlasters (которые мы бы не рекомендовали) это не проблема, но для профессиональных устройств от других производителей может стать таковой. Исключение — RME, так как автор *Ardour* пользуется одной из этих карт.

Второй тип интерфейса — Firewire. Такие устройства часто дают наилучшую производительность при многорожечной записи с высокими битрейтами и частотами дискретизации. Но Firewire выходит из моды, и драйверы FFADO, необходимые для работы этих устройств в Linux, применимы только к небольшому набору устройств. К тому же устройства на Firewire обычно недешевы.

Третий тип интерфейса — скромный USB. Такие карты, безусловно, наиболее распространены, и благодаря стандартной природе реализации звука через USB, многие карты будут просто работать в Linux без всяких дополнительных настроек (но учтите: есть исключения). Просто подключите их, а ядро сделает все остальное. Так как эти устройства скорее всего будут работать с системой Jack, а не с главным аудиовыходом, нужно только выбрать корректные входное и выходное устройства на странице настроек QJackCtl, перед этим выбрав Duplex [Дуплексный] в настройках звука.

Сейчас наше любимое устройство — Alesis iO4. Оно подключается через USB и предоставляет четыре независимых входа и выхода. Четыре входа можно настроить для гитар, микрофонов или как обыкновенные линейные входы с фантомным питанием или без него. На всех входах есть светодиодные измерители уровня сигнала, а также можно управлять усилением входного сигнала.



В окнах подключения *Ardour* аппаратные и программные входы подсоединяются к дорожкам и шинам.

ПИСАТЬ МУЗЫКУ

Кнопка direct/USB [напрямую/через USB] позволяет просматривать входной сигнал, не дожидаясь его обработки компьютером, что необходимо, если вы хотите воспользоваться наушниками во время записи. Единственный недостаток Alesis iO4 в том, что если включить поддержку 24-битного звука, вы сможете пользоваться только двумя входами одновременно, но это ограничение ширины канала USB, а не звуковой карты.

Подключаемся

В *Ardour* используется звуковая подсистема под названием Jack. Jack позволяет любым Jack-совместимым приложениям легко обмениваться данными друг с другом. Например, можно направить вывод синтезатора или драм-машины прямо в *Ardour*, не заботясь о разделении приложениями общего выходного звукового устройства. К сожалению, в Jack нелегко использовать это устройство совместно с PulseAudio, звуковой подсистемой, ныне применяемой в большинстве дистрибутивов. Чтобы это обойти, лучше всего воспользоваться небольшой утилитой под названием *pasuspender*. Она временно отключает PulseAudio, пока звуковое устройство используется другим процессом, и должна быть уже установлена, а для ее настройки, как и для настройки самого Jack, лучше всего воспользоваться графической программой под названием *QJackCtl*. Как и *Ardour*, она имеется в менеджере пакетов вашего дистрибутива, а запустив ее, вы увидите небольшое окно, похожее на окно управления CD-проигрывателем.

Запустив *QJackCtl*, нажмите кнопку Setup [Параметры], и откроется страница настроек. Для начала измените поле Server Prefix [Префикс сервера], включив в него *pasuspender*. Для этого укажите **pasuspender --** перед командой, используемой для запуска сервера Jack (обычно **jackd**). Обязательно укажите два дефиса. Остальные настройки можно не менять, но лучше задать их в соответствии с вашей картой. Измените Audio [Звук] на Duplex [Дуплексный], так как это позволит записывать и воспроизводить звук одновременно, и вручную выберите входное и выходное устройство из выпадающих меню справа от этих полей. Еще одна важная настройка на данной панели — задержка звука для вашей звуковой карты. Задержка — это время, требуемое карте на обработку звука; оно зависит от возможностей карты и используемых фреймов, буферов и частот дискретизации. Оно повлияет и на задержку перед тем, как вы услышите входной сигнал, и на задержку между *Ardour* и вашими колонками или наушниками.

Увы, лучший способ поиска наименьшей задержки для вашей карты — путь проб и ошибок. Например, для нашей Alesis нам удалось получить отличную производительность при соотношениях Frames/Period [Фреймов в периоде] 256 и Periods/Buffer [Периодов в буфере] 2 на странице настроек *QJackCtl*. Это дало нам задержку в 11,6 мс, которая почти незаметна при мониторинге в *Ardour*, и большинство карт должно дать похожие результаты. Если вы задали слишком маленькие значения параметров, то в окне *QJackCtl* начнут появляться задержки, а также искажения звука, и вам понадобится увеличить значения и проверить их при высокой загрузке процессора.



» Автоматизация поможет записывать и устанавливать перемещения ползунков во времени, как и многие другие параметры.

Есть и другой вариант. Если вы хотите просто попробовать *Ardour*, не тратя времени на *QJackCtl*, и особенно если у вас встроенная звуковая карта, можно просто запустить *Ardour*, и она сама постарается создать подходящие настройки в Jack для вашей карты. Тогда можно вообще забыть о Jack, но наш опыт научил нас, что лучше смотреть опасности в лицо и настраивать Jack напрямую, особенно если запущена PulseAudio.

Запускаем Ardour

При первом запуске программы появится маленькое окно, спрашивающее, нужно ли создать новую сессию или открыть существующую. Выберите первый вариант и введите имя сессии и ее расположение. Это расположение будет использоваться для хранения всех аудиоданных проекта. Один час звука в формате стерео обычно занимает около 650 МБ пространства на диске, и это стоит принять во внимание, если вы сохраняете сессию в домашний каталог, где место может быть ограничено. После нажатия кнопки Open [Открыть] запустится сама программа.

На первый взгляд *Ardour* выглядит устрашающе — это факт. Она никак не пытается поощрить новичков или тех, кто ждет нечто похожее на *Garageband* от Apple. Но после ознакомления с интерфейсом все будет не так уж трудно. Скрыто совсем немного параметров, а если вы считаете *Ardour* клиентом для виртуальной студии звукозаписи, это тоже логично. В верхней части окна находятся глобальные элементы управления проекта: перемотка вперед и назад, воспроизведение и запись, а также включение метронома и точки входа и выхода записи. Прямо под ними находятся треки с маркерами, которые могут изменяться с течением времени — например, темп и размер фрагмента или расположение различных треков на мастере компакт-диска.

Ниже находится самая важная часть программы — в ней размещаются аудиотреки. По умолчанию в новом проекте их нет, есть только «главная шина [master bus]», которую вы видите сверху. Любые треки, которые вы добавите, будут пропущены через эту главную шину, чтобы можно было сжать окончательный вариант звука или создать микс из всего проекта. Но, как вы скоро

Скорая помощь

Для наилучшей производительности своей звуковой карты установите версию ядра реального времени [realtime]. Так вы получите меньшую задержку и повысите стабильность звука.

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

убедитесь, Ardour — полностью модульная программа, поэтому звук с трека не обязательно пропускать через главную шину — это лишь маршрут по умолчанию.

Для создания нового трека щелкните правой кнопкой мыши под главной шиной. Откроется окно Add Track or Bus [Добавить трек или шину], в верхней части которого можно выбрать тип трека. В Ardour есть три типа треков...

Аудиотреки

Это самый распространенный тип треков в Ardour. Они аналогичны звуковым трекам в редакторах вроде Audacity, и могут содержать импортированный звук или звук, записанный непосредственно в Ardour. В большинстве проектов аудиотреки состоят из одного (моно) или двух (стерео) каналов, но Ardour умеет работать и с большим количеством каналов для получения кругового звука и других задач. Треки можно редактировать, разбивая их на блоки, плавно уменьшая или увеличивая громкость в одном или нескольких блоках, а также перетаскивая дорожки внутри редактора. В отличие от Audacity все это редактирование неразрушающее — то есть исходные звуковые файлы не затрагиваются. Это можно изменить в меню Track Mode [Режим трека]. По умолчанию используется обычный режим [Normal], в нем при записи нового звука поверх другого всегда создается новый файл. Режим Без слоев [Non-layered] все еще является неразрушающим, но блоки под новой записью обрезаются, чтобы освободить место для нового звука. В режиме Запись [Tape] новый звук переписывает данные в исходных аудиофайлах, поэтому мы не рекомендуем им пользоваться.

Треки MIDI

Эти треки — главное дополнение к последнему релизу. Они позволяют Ardour управлять устройствами и программами MIDI. MIDI — старый протокол, часто используемый для отправки данных нот синтезаторам, но он также может управлять многими другими внешними устройствами, а также внутренними программными инструментами. Треки MIDI позволяют создавать и редактировать данные MIDI напрямую рядом с аудиотреками. В окне Инструмент [Instrument] в окне добавления трека можно выбрать любые программные инструменты LSSI и эффекты, уже установленные в вашей системе. Так Ardour подключит виртуальные устройства MIDI автоматически и не потребует от вас



» Благодаря неразрушающему редактированию можно разрезать, перемещать и уменьшать громкость блоков звука, не боясь повредить исходные файлы.

подключать их вручную. Для внешних устройств или других программных инструментов подключение придется выполнить вручную после создания трека MIDI — скоро мы покажем, как.

Треки аудио + MIDI

В других цифровых рабочих станциях для обработки звука, таких как Cubase, треки аудио + MIDI позволяют легко интегрировать в рабочее окружение программные синтезаторы. Они позволяют создавать данные MIDI для отправки на синтезатор, а также и управлять звуком. В Ardour треки аудио + MIDI имеют другое назначение. Они предназначены для плагинов, которые принимают как данные MIDI, так и звуковые данные; например, для плагина сдвига тона [pitch shifter], который на основе данных MIDI выбирает тон, или плагина вокодера. Поэтому они чуть менее полезны.

Шины

В мире аналоговой аудиозаписи шины используются для группировки аудиосигналов с других каналов для обработки или управления. Например, с помощью шины можно к группе звуковых эффектов добавить реверберацию. Звук отправляется на шину с помощью т.н. Aux Send. Шины можно добавить на каждый канал

Скорая помощь

В микшере можно добавлять эффекты в режиме префейдера, перетаскивая их над устройством Fader [Фейдер] в разделе вставки.

Записываем виртуальный синтезатор



1 Устанавливаем синтезатор

Этот алгоритм должен подойти для любого синтезатора (мы воспользовались Bristol). Установите выбранный синтезатор и запустите его с командной строки: `startBristol -audio jack -arp2600 -midi jack`.

2 Настраиваем треки

Добавьте аудиотрек, а затем трек MIDI. В менеджере подключения звука [Audio Connection Manager] подключите Bristol из источника [Source] Другие [Other] к треку 1 Ardour [Ardour Tracks Audio 1]. Подключите Выход 1 MIDI [MIDI 1 Out] к Другим/Bristol [Other/Bristol] в менеджере подключений MIDI [MIDI Connection manager].

3 Редактируем треки

Увеличьте масштаб трека MIDI, и вы сможете добавлять блоки и примечания карандашом. Чтобы записать музыку в аудиотрек, запишите ее и нажмите Play Record [Воспроизвести запись] на главной панели инструментов.

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

в микшере *Ardour*, и они позволяют управлять количеством звука, отправляемого на шину, а также поместить звук до или после фейдера. При использовании префейдера на звук не будет влиять громкость фейдера, а при использовании постфейдера она будет пропорциональна громкости фейдера.

Связываем каналы

Чтобы понять, как работает *Ardour* и как сделать свои первые записи, нужно заглянуть внутрь и рассмотреть взаимосвязь звука, MIDI и шин. Это возможно благодаря тому, что всем этим управляет Jack, а в *QJackCtl* можно видеть, что происходит. Если нажать Connect [Подключить] и убедиться, что выбрана вкладка Audio [Звук], то вы увидите, как *Ardour* подключается к нашей звуковой системе. Треки, шины и другие каналы генерации звука перечислены в левой панели, а их места назначения — справа. Например, главная шина должна быть подключена к системным выходам — в противном случае вы ничего не услышите! Аналогично, для записи входа со звукового устройства нужно подключить системный выход звукового устройства к аудиовходу канала *Ardour* (по умолчанию Audio 1/audio_in 1). Это не происходит автоматически, но в *Ardour* есть собственный исчерпывающий интерфейс для подключения, и вам не придется обращаться к *QJackCtl*.

Желая подключить физический вход к новому аудиоканалу в *Ardour*, выберите меню Window [Окно], а затем Audio Connection Manager [Менеджер аудиоподключений]. Эти окна подключений в *Ardour* эквивалентны окну Connect [Подключение] в *QJackCtl*, с той разницей, что Source [Источник] и Destination [Место назначения] находятся на других осях, а вместо виртуальных кабелей используется сетка. Источники [Sources], как источники на вашем аудиоинтерфейсе, находятся сверху на вкладке Hardware [Оборудование]. Если входы не определены, например, если в Jack настроено воспроизведение, они не появятся. Чтобы подключить вход к треку для записи, кликните на Audio Tracks [Аудиотреки] в списке назначения и затем на ячейке матрицы, чтобы создать соединение. Если подключить к одному каналу несколько входов, сигнал будет смикширован. Вернувшись в главное окно, вы увидите, как указатель уровня сигнала прыгает вверх-вниз, отражая уровень сигнала на входе (записать который можно, щелкнув по красной кнопке Record Enable [Включить запись] у трека), а затем большие кнопки записи и воспроизведения на главной панели инструментов в верхней части окна.

Редактирование и микширование

По мере записи звука вы увидите, как на дорожке будут появляться блоки записи. Если вам повезет, уровень сигнала будет достаточным, чтобы вы видели форму записанного сигнала. Она

должна быть вам знакома, если вы пользовались звуковыми редакторами вроде *Audacity* — пики и провалы представляют амплитуду сигнала. Если форма сигнала плоская, взгляните на настройки микшера ALSA или подключите внешний усилитель. Для удобного редактирования трека можно воспользоваться элементами интерфейса трека *Ardour*. Например, чтобы удлинить трек, перетащите его контуры, а чтобы увеличить масштаб части трека, воспользуйтесь колесиком мыши. В нижней части окна находится обзор проекта, с помощью которого можно легко перемещаться по аудиотрекам, так как это эластичное окно динамически изменяет размер треков при перемещении по ним.

Процесс редактирования зависит от материала, с которым вы работаете, но обычно включает прослушивание (нажмите пробел для воспроизведения), паузу (снова нажмите пробел), изменения и дальнейшее воспроизведение. Во время воспроизведения вы увидите, как по трекам движется курсор. Он обозначает положение воспроизведения. Его можно переместить вручную, проведя мышью по маркеру в верхней части окна. Режим редактирования можно выбрать на панели инструментов; самый распространенный — с иконкой «указующая рука». В этом режиме можно разрезать видео под курсором, нажав S, и переместить фрагменты трека с помощью курсора. Точки редактирования также можно изменить еще одним перетаскиванием, а перетаскивание якоря в верхней части этих разделов, можно создать затухание звука. Чтобы разрезать несколько блоков, просто выберите их сразу — тот же прием работает при изменении размеров нескольких блоков. Так можно редактировать и строить свой трек совершенно так же, как в любой программе для обработки цифрового звука.

Наконец, завершив редактирование, можно экспортировать и смикшировать весь проект в один файл с помощью пункта меню Export [Экспорт]. Но сначала убедитесь, что маркеры Start [Начало] и End [Конец] указывают на начало и конец вашего проекта. Окно экспорта выглядит сложно, но выбрав пресет — а пресеты есть для самых популярных выходных форматов — можно избежать ввода такой прорыва параметров вручную. Вам может понадобиться выбрать главную шину, открыв параметры Time Span And Channel Options [Параметры временного диапазона и канала], потому что выходные данные именно с этой шины должны попасть в файл. LXF



► Alsis iO4 подключается через USB и не требует дополнительных драйверов, поэтому прекрасно подходит для записи предварительно усиленного сигнала с четырех входов.

Микшер Ardour

Одно из самых важных окон *Ardour* — микшер. Он находится в меню Window [Окно] и содержит все треки и шины проекта, расположенные так же, как они могли бы быть расположены на настоящем микшере в студии звукозаписи. Задача микшера — позволить вам изменять уровни сигналов, а также добавлять эффекты и обрабатывать звук. Уровни легко изменить с помощью фейдера, но если эти изменения должны быть автоматизированы с течением времени, вам нужно выбрать «M» под фейдером. При этом откроется маленькое выпадающее меню, где нужно выбрать Write [Запись]. Теперь любые изменения в фейдере будут записаны в трек автоматизации. Чтобы уровень менялся автоматически, измените этот режим на Read [Чтение]. Желая

просмотреть эти данные, вернитесь обратно в окно просмотра, щелкните правой кнопкой мыши на треке, выберите меню Automation [Автоматизация], затем Fader [Фейдер].

В микшере также можно добавить эффекты. Для этого щелкните правой кнопкой мыши по черной области над ползунком и выберите эффекты, которые хотите добавить, с помощью менеджера плагинов.

Ardour очень придирчив в отношении входов и выходов, поэтому вам нужно убедиться, что выбранный эффект настроен для обработки каналов вашего трека. Здесь также можно добавить Aux Sends, чтобы добавить некоторый звук канала в шину.



► Микшер используется для управления выходами в реальном времени, включая относительные уровни, и для добавления эффектов.

Детские игры:

Хотите ограничить общение своих детей с компьютером, не стоя у них над душой? Нейл Ботвик создает скрипт командного процессора, который сделает это за вас...



Наш эксперт

У Нейла Ботвика по компьютеру в каждой комнате, но по соображениям безопасности он ни за что не скажет вам, где находится центральный сервер.



В LXFI66 Майк Сондерс уже рассказывал о том, как защитить ваших детей от темной стороны Интернета — и от них самих. Ну, а если вы также хотите контролировать, когда и как долго они пользуются компьютером? Например, выставить запрет на сидение по ночам, или лимит на Facebook не более 2 часов. *Gnome Nanny* [Нянюшка Gnome], которая кое-с чем из этого справлялась, больше не разрабатывается и несовместима с Gnome 3. Та же участь постигла *Timekpr* (хотя разработчик планирует воскресить его в этом году).

Стало быть, сейчас в Linux нет простого способа типа наведи-и-щелкни, чтобы ввести такие ограничения. Но не все потеряно — это можно сделать; надо лишь немного утрудить ручки.

PAM

Ограничение использования компьютера по времени осуществляется при помощи PAM (Pluggable Authentication Module, Подключаемые модули аутентификации). Прежде чем перейти к делу, разберемся в общих чертах, что это и как работает. Раньше программы работали непосредственно на уровне аутентификации

пользователя, обычно выполняя ее через файл `/etc/passwd`. А значит, при введении нового способа аутентификации каждую программу приходилось перекодировать под него. PAM предоставляет для этой процедуры отдельный уровень. Программа обращается к PAM, а тот занимается собственно аутентификацией. По стандартному ли вводу логина/пароля, или с помощью дактилоскопического анализа, или грубых жестов в адрес Kinect — программе не важно. Все это делает PAM, а затем сообщает программе, прошел ли пользователь проверку. Выглядит процесс аутентификации вроде бы более трудоемким; так оно и есть, но иногда лучше лишний раз перестраховаться. Настройки PAM хранятся в ряде файлов в `/etc/pam.d`, и имя каждого файла связано с соответствующей службой:

```
auth required pam_securetty.so
auth include system-local-login
account include system-local-login
password include system-local-login
session include system-local-login
```

Первый элемент в строке указывает на действие PAM, за которое он отвечает, второй — на то, обязательно [required] его выполнение или достаточно [sufficient], третий — на имя модуля PAM, применяемого для аутентификации. То есть первая строка обозначает, что действие **auth** осуществляется при помощи модуля *pam_securetty*, и статус его успеха — **required**. Если модуль не проходит аутентификацию, процесс завершается, а если проходит, PAM переходит к следующему правилу. Флажок **sufficient** указывает, что данная проверка полностью обеспечивает аутентификацию и PAM возвращает результат «выполнено», если она завершается успешно. Есть и другие опции (описанные на ман-страницах), в том числе **include**. Она добавляет выполнение условий из другого файла, так что вторая строка дублирует правила **auth** из `/etc/pam.d/system-local-login`, который нам вскоре пригодится. Заметим, что в некоторых дистрибутивах вместо него используется формат **@include common-account**. Но это мелкое отличие для нас неважно. Важно заглянуть в файлы из `/etc/pam.d`, чтобы понять, какие из них нужно редактировать.

Время зайти

Возвращаясь к нашей теме, один из модулей PAM называется *time* и контролирует аутентификацию пользователя на основе даты и времени. Это значит, что вы можете запретить вход в систему, допустим, позднее 8 вечера по будням и 9 по выходным, включая другие, более тонкие варианты. Делается это за два шага:

Лучше просвещать, чем запрещать

Лучший способ общения с детьми — да и с людьми вообще — просвещать, а не налагать запреты. Приведенные здесь настройки доступа — не панацея,

они не отменяют присмотра и обучения. Ваши дети должны сами понимать, что можно и чего нельзя, а контроль должен лишь укреплять это. Модуль *pam_time*

записывает все неудавшиеся попытки войти в **syslog**, так что вы можете проверить, ведут ли дети себя хорошо или их удерживают лишь ваши правила.

Потехе — час

подключение модуля и его настройка. Следуя нашей логике, мы в первую очередь остановимся на подключении, но на практике лучше начать с настроек.

Как и вышеупомянутый файл `system-local-login`, так и файлы PAM для KDE, `Gdm` и `Lightdm` (или любого другого настольного менеджера паролей) — включают `system-auth` (или `common-account` — на производных Ubuntu). Определив настройки времени в этом файле, мы применим их ко всем методам входа в систему: графическому, консольному или SSH. Отредактируйте файл, добавив в его начало строки

```
# enable time based controls
account required pam_time.so
```

В начало они ставятся для того, чтобы предыдущая строка не признала аутентификацию до того, как в дело вступит `pam_time`. А так войти в систему можно будет не раньше, чем PAM сверит данные о пользователях и времени, когда им разрешен вход. Список расположен в файле `/etc/security/time.conf`. Вот простой пример конфигурации:

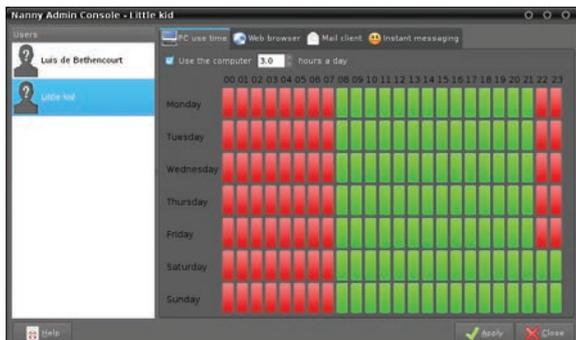
```
* ; * ; jimmy ; SaSu0800-2100
```

Каждая строка включает четыре поля, разделенных точкой с запятой: первое — имя соответствующей службы PAM, второе — список TTY (имен в логическом терминале), которых касается действие; мы в данном случае решили применить правило ко всем. В третьем поле указывается имя (имена) пользователя (-лей) — мы использовали только одно, но можно ввести и несколько в виде логической строки (`jim | bob` означает `jim` или `bob`). Интересно четвертое поле — это значения времени. Здесь указаны дни недели, с использованием двухбуквенных сокращений, и временные промежутки в формате 24 часа. В нашем примере мы позволяем Джимми входить в систему по субботам и воскресеньям с 8 утра до 9 вечера.

Если вторая отметка времени раньше, чем первая, это понимается как переход на следующий день, и чтобы запретить доступ по ночам в будни, можно использовать

```
* ; * ; jimmy ; !SuMoTuWeTh2100-0800
```

Заметим, что `!` обозначает «не» (`!` применимо и к именам пользователей), то есть Джимми не сможет войти в систему в указанное время.



» **Gnome Nanny** и **Timekpr** позволяли сделать то же самое в графическом интерфейсе.

```

TW system-auth (Modified) Row 4 Col 1 9:55 Ctrl-K H for help
# enable time based controls
account required pam_time.so

# log all logins to syslog
account required pam_warn.so

# run scripts on session open and close
session optional pam_script.so

auth required pam_tally2.so deny=8 unlock_time=20 silent
auth required pam_env.so
auth required pam_unix.so try_first_pass likeauth nullok
auth optional pam_permit.so

account required pam_unix.so
account optional pam_permit.so

password required pam_cracklib.so difok=2 minlen=8 dcredit=2 ocredit
password required pam_unix.so try_first_pass use_authok nullok sha5
password optional pam_permit.so

session required pam_limits.so
session required pam_env.so
session required pam_unix.so
session optional pam_permit.so
  
```

» Редактирование файла PAM `system-auth` для включения настроек времени и скриптов, а также журналирования.

Есть установленные сокращения для дней: `Wk` значит будни, `Wd` — выходные (осторожно, их легко спутать), а `Al` — это все дни. Двойное упоминание дня отменяет его, то есть `AlSa` — это все дни, кроме субботы. Будет разумно поместить

```
* ; * ; yourusername ; Al000-2400
```

в начало файла, чтобы не заблокировать себе вход в систему из-за опечатки. PAM — это не демон, его не нужно перезагружать или передавать ему сигнал после внесения изменения в настройки. Когда вы сохраните файл, он будет активен при следующем запуске PAM.

Время уйти

PAM может контролировать вход в систему, но не может прерывать текущий сеанс; однако это легко сделать с помощью `Cron`. Следующий скрипт дает пользователю KDE (будет вам и версия Gnome) пять минут, чтобы завершить сеанс или выключить компьютер, или это делается принудительно:

```
#!/bin/sh
THEUSER="username"
[[ -z "$(w -h $THEUSER)" ]] && exit
DISPLAY=":0" kdialog --msgbox "Пора спать!\n\nВыключение через 5 минут." & sleep 5m
source /home/$THEUSER/.dbus/session-bus/$(cat /var/lib/dbus/machine-id)-0
qdbus --address $DBUS_SESSION_BUS_ADDRESS org.kde.ksmserver /KSMServer logout 0 3 3
```

Скрипт запускается при помощи `w` (да, это команда), проверяет наличие пользователя в системе, затем выводит на экран предупреждение, ожидает пять минут и затем отправляет широкое сообщение службам KDE D-BUS. Файл исходного кода используется для распознавания файла, содержащего адрес пользовательского сеанса D-BUS и определения указанных в нем переменных. Необходимо, чтобы он запустился за пять минут до отключения, и это задача `Cron`. Запускаться он должен от имени того же пользователя, поэтому добавим в `crontab`

```
sudo crontab -e -u username
```

Вот как запланировать выход в 9 вечера с воскресенья до четверга и с 10 по выходным:

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

```
55 20 * * 0-4 /path/to/script
55 21 * * 5,6 /path/to/script
```

Нам не надо, чтобы пользователь мог изменить эти данные в *Cron*, поэтому добавим его имя в **/etc/cron.deny** и убедимся, что этот скрипт не сможет быть им перезаписан или не находится в директории, которая может быть перезаписана (ведь если доступ к директории у вас есть, вы можете удалить из нее файлы и переименовывать их, пусть и не внося изменения внутри). Хороший вариант расположения — **/usr/local/bin**.

Как и обещано, представим вариант для Gnome и Unity. Команда **gnome-session-quit** существует в рабочем сеансе, но выход из системы должен подтверждаться самим пользователем, так что обратимся к D-BUS. Вот версия для Gnome:

```
#!/bin/sh
THEUSER="username"
[[ -z "$(w -h $THEUSER)" ]] && exit
DISPLAY=:0 zenity --info --text="Пора спать\n\nВыключение через 5 минут." & sleep 5m
source /home/$THEUSER/dbus/session-bus/$(cat /var/lib/dbus/machine-id)-0
qdbus --address $DBUS_SESSION_BUS_ADDRESS org.gnome.SessionManager /org/gnome/SessionManager org.gnome.SessionManager.RequestShutdown
```

Вместо *Kdialog* мы использовали для отображения сообщения *Zenity* и изменили вызов D-BUS. Если у вас другой рабочий стол, вы тоже можете использовать D-BUS, или же вам придется действовать жестко и убивать X-процесс. Можно использовать **notify-send** для создания всплывающего уведомления в строке состояния, каким бы окружением вы ни пользовались — хотя такие сообщения все же менее заметны, чем в виде окна:

```
DISPLAY=:0 notify-send "Пора спать" "Выключение через 5 минут."
```

Некоторые умельцы, узнав, что вы заблокировали их учетную запись, могут попробовать войти через вашу, чтобы обойти настройки. Вы, конечно, выбрали надежный пароль, желательно

не имя того ребенка, которого вы хотите проучить; впрочем, навряд ли их это остановит. Зато это делает модуль *pam_tally2*. После нескольких неудачных попыток входа он заблокирует учетную запись: либо на определенный срок, либо пока вы не решите ее разблокировать. Активируйте и настройте модуль в **system-auth** при помощи строки

```
auth required pam_tally2.so deny=5 unlock_time=7200
```

Это заблокирует все варианты входа, кроме *root*, на два часа (7200 секунд) после пяти неудачных попыток. Если вы хотите, чтобы *pam_tally2* контролировал и *root*, добавьте в настройки **even_deny_root**. Если время разблокирования не указано, учетная запись становится условно активной. Вы можете увидеть статус пользователя через

```
sudo pam_tally2 --user jimmy
```

и разблокировать его учетную запись без таймаута, а если таковой не определен, то с помощью

```
sudo pam_tally2 --user jimmy --reset
```

Имейте в виду, что при такой блокировке вы не сможете войти в систему ни под своим именем, ни от имени *root*, но сможете через SSH, если был установлен ключ аутентификации (в этом случае процесс идет внутренне, без вмешательства PAM). Убедитесь, что этот вариант вам доступен, прежде чем установить такие ограничения. Кроме того, запароленный вход через SSH может стать дополнительной мерой подстраховки от вторжения хулиганов-скриптописцев.

Бонусы PAM

Модулей в PAM и так полно, но можно добыть и больше — как правило, через репозитории вашего дистрибутива. Модуль *pam_script* ограничивает время использования компьютера, но многие, возможно, захотят ограничить и продолжительность сеанса. Для этого установите *pam_script* как обычно и добавьте такую строку в файл **system-auth**:

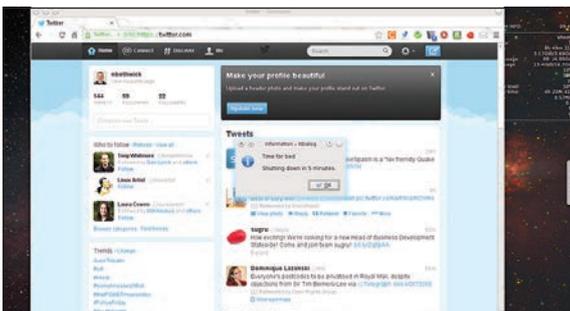
```
session optional pam_script.so
```

При каждом закрытии и открытии сеанса *pam_script* будет запускать любые скрипты из **/etc/security/pam-script/pam-script.d**, заканчивающиеся на **_ses_open** либо **_ses_close**. Флаг **optional** обозначает, будет ли аутентификация блокироваться, если скрипт не выполнен. Помните тот, что выводит пользователя из системы, в *Cron*? Мы можем создать *pam_script*, который будет запускать такой же через определенное время после входа. Добавьте в **/etc/security/pam-script/pam-script.d/timeout_ses_open**

```
#!/bin/sh
echo "sudo -u $PAM_USER /path/to/our/logoutscript" | at -q z now + 1 hour
```

Демон *at* запускает команду в заданное время однократно — не как в *Cron*. Данная команда запустит скрипт через час после

➤ **Пора спать!**
Небольшой скрипт в паре с *Cron* заставят малыша выключить компьютер — хочет он того или нет.



Ограничение прав администрирования

Чтобы менять эти настройки, вам нужен доступ в *root*, через **su** или **sudo**. Разумеется, те, кого вы пытаетесь ограничить, не должны обладать такими же правами, иначе они легко отменят ваши нововведения. При использовании **su** решение простое: не выдавать пароль *root*. А для большинства дистрибутивов с **sudo** запускать программы от имени *root* может первый созданный пользователь. Если это компьютер вашего ребенка и права администрирования принадлежат ему, войдите

под его именем, создайте другого пользователя и добавьте ему привилегии. В дистрибутивах это делается по-разному; поищите в **/etc/sudoers** строку вида

```
%groupname ALL=(ALL) ALL
```

Группа обычно называется **wheel** или **adm**. Добавьте туда своего пользователя, затем выйдите и войдите, чтобы убедиться, что вы можете использовать **sudo**, а после этого удалите пользователя своего ребенка из этой группы.

Это радикальный метод, из категории «все или ничего». **Sudo** дает большую гибкость. Отредактируйте файл **sudoers** при помощи *visudo* (никогда не делайте этого напрямую, если только вы не сидите с *live CD*, восстанавливая настройки после того, как сами себя заблокировали) и добавьте нечто вроде

```
username ALL = /usr/bin/apt-get, /usr/bin/apt-cache, /usr/bin/synaptic
```

чтобы разрешить ребенку устанавливать программы, но не более того.

➤ **Пропустили номер?** Узнайте на с. 108, как получить его прямо сейчас.

текущего момента и добавит его в очередь, называемую **z**. Демон **at** поддерживает множество очередей с однобуквенными именами. По умолчанию используется **a**; мы здесь используем **z**, но это может быть любая очередь, не упоминаемая где-либо еще, поскольку по завершении сеанса значения удаляются. Через час после входа в систему скрипт запускается как раньше. Естественно, мы не хотим, чтобы он запускался, когда пользователя и так нет в системе, поэтому добавим следующее в **/etc/security/pam-script/pam-script.d/timeout_ses_close**:

```
#!/bin/sh
atrm $(atq -q z | awk '{print $1}')
```

Awk добывает номера задач из списка ожидающих в **atq** и передает их **atrm** для удаления. Это гарантирует, что очередь будет использоваться только для одного пользователя, как и должно быть в данной ситуации, иначе вам придется проверять имена пользователей.

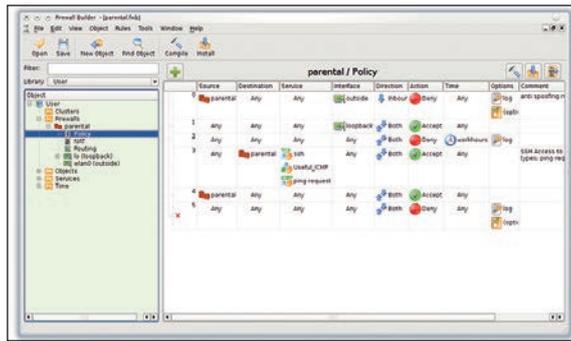
Интернет-активность

Контроль использования компьютера — это только начало; как насчет ограничения действий? Некоторые роутеры позволяют ограничить доступ конкретных машин в Интернет по времени, сервису или URL — или их комбинации. Открытая ОС для роутеров DD-WRT, которая подходит для многих популярных моделей, а на некоторых уже установлена по умолчанию, предлагает гибкие варианты настройки. Можно, к примеру, закрыть доступ к социальным сетям по URL, а также службам сообщений и *Skype*, не ограничивая поиск в учебных целях. Детали вы найдете в справке по вашему роутеру, но если в нем таких фильтров нет, можно использовать системный брандмауэр.

С помощью модуля **xt_time** вы можете добавить в **Iptables** правило блокировать получение и отправку пакетов, не входящих в локальную сеть, вот так:

```
iptables -A OUTPUT ! -d 192.168.1.0/24 -m time --timestart 21:00
--timestop 08:00 --weekdays Sun,Mon,Tue,Wed,Thu -j DROP
iptables -A INPUT ! -d 192.168.1.0/24 -m time --timestart 21:00
--timestop 08:00 --weekdays Sun,Mon,Tue,Wed,Thu -j DROP
```

Учтите, что в **Iptables** по умолчанию используются настройки UTC, а не местного времени, то есть их нужно будет подкорректировать. Также вам придется обновлять эти правила дважды в год с переходом на летнее время и обратно. Чтобы этого избежать, можно добавить опцию **--kerneltz** к вышеприведенному правилу и затем использовать местное время. Можно также фильтровать исходящие пакеты по отправителю, замкнув цепочку **OUTPUT** только на одном пользователе:



➤ Если вы — не гуру **iptables**, вам будет проще и безопаснее воспользоваться одним из графических межсетевых экранов. **Firewall Builder** можно использовать для создания и установки правил для другого компьютера вашей сети.

```
iptables -A OUTPUT ! -d 192.168.1.0/24 -m time --timestart 21:00
--timestop 08:00 --weekdays Sun,Mon,Tue,Wed,Thu -m owner
--uid-owner jimmy -j DROP
```

Написание правил брандмауэра — задача непростая: добавить единичное правило к каждой цепочке, как показано выше, с чистого листа, может, и легко, но при изменении существующих правил с добавлением новых функций результат может быть непредсказуем, если вы не знаете точно, что делаете. Если вы уже пользуетесь брандмауэром, надежнее будет воспользоваться одним из существующих инструментов настройки, будь то графическая программа вроде *Firewall Builder* или что-то наподобие *Shorewall*. Эти приложения проверяют разумность новых настроек, прежде чем их применять.

По следу

PAM фиксирует все неудачные попытки входа в систему в **syslog**, но можно заставить его записывать вообще все попытки, добавив такую строку в файл **system-auth** или **common-account**:

```
account required pam_warn.so
```

Модуль **pam_warn** на аутентификацию не влияет, но записывает попытку в системный журнал. Еще один полезный источник информации — **ConsoleKit** в **/var/log/ConsoleKit/history**. Здесь содержатся данные не только о входе и выходе пользователей из системы, но и о переключении сеансов, как, например, переход от **X** к виртуальной консоли через **Ctrl+Alt+F1**. Есть там и сведения о входе через **SSH**, так что понадобится немножко магии **grep**, чтобы их извлечь.

```
grep "session-x11-display=:0".*session-unix-user=1001 /var/log/ConsoleKit/history
```

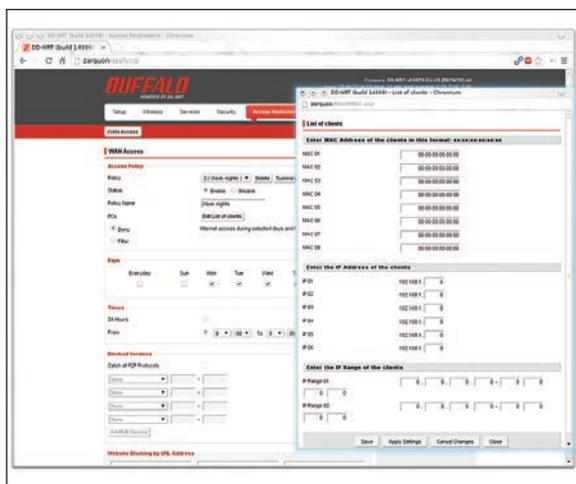
Указав цифровой ID пользователя, вы увидите все его входы и выходы в **X**, вот в таком виде:

```
1366128164.765 type=SEAT_SESSION_ADDED : seat-id='Seat1'
session-id='Session65' session-type=' session-x11-display=:0'
session-x11-display-device='/dev/tty7' session-display-device='
session-remote-host-name='session-is-local=TRUE session-unix-
user=1000 session-creation-time='2013-04-16T16:02:44.761963Z'
1366285984.660 type=SEAT_SESSION_REMOVED : seat-
id='Seat1' session-id='Session65' session-type='
session-x11-display=:0' session-x11-display-
device='/dev/tty7' session-display-device=' session-remote-
host-name='session-is-local=TRUE session-unix-user=1000
session-creation-time='2013-04-16T16:02:44.761963Z'
```

Входы обозначаются как **type=SEAT_SESSION_ADDED**, а выходы — **type=SEAT_SESSION_REMOVED**. Каждая запись включает время начала сеанса; если же вам необходимо знать время окончания, придется конвертировать значение времени в начале строки из стандарта Unix (в секундах, прошедших с 1 января 1970 года) в привычный формат, при помощи команды:

```
date --date='@1366285984.660'
```

которая показывает, что сеанс завершился в четверг, 18 апреля, в 12:53:04. Можете совместить все это, если вас увлекает магия **Awk**; здесь мы даем вам поупражняться самостоятельно. **LXF**



➤ Многие роутеры поддерживают настройку доступа по конкретным сайтам и времени. **DD-WRT** — хороший и открытый вариант!



Erlang: ОПЯТЬ

Андрей Ушаков возобновляет практикум по созданию многозадачных версий функций.



Наш эксперт

Андрей Ушаков активно приближает тот день, когда функциональные языки станут мейнстримом.

Мы в очередной раз продолжаем в качестве практикума по многозадачности решать задачу по созданию многозадачных версий функций **map** и **reduce**. Но теперь мы создадим последний вариант этих функций и остановимся, чтобы оценить то, что мы сделали (правда, оценить наши результаты мы сможем только на следующем уроке), и пойти дальше.

Давайте посмотрим, на чем мы остановились в прошлый раз. На прошлом уроке мы переработали подход к созданию и распределению заданий для рабочих процессов при реализации очередных многозадачных версий функций **map** и **reduce**. До переработки подхода к созданию и распределению заданий все исходные данные мы сразу же разбивали на порции и рассылали рабочим процессам. После переработки подхода мы стали создавать порции и отсылать их рабочим процессам только по требованию — в тот момент, когда рабочий процесс закончил обрабатывать некоторую предыдущую порцию данных и отослал результаты обработки главному процессу.

Понятно, что при новом подходе мы не создаем одновременно большое количество новых объектов (порций данных, которые являются кортежем из двух элементов: индекс порции и собственно данные порции) и не отсылаем одновременно большое количество данных рабочим процессам. Однако следует заметить, что следующее фундаментальное свойство рабочих процессов сохраняется при переходе от старого подхода к новому: все рабочие процессы являются эквивалентными относительно распределения задач между ними. Это означает, что как только какой-либо рабочий процесс освободился, новое задание на обработку мы дадим именно этому процессу.

Здесь у читателя может возникнуть вопрос: насколько хорошо, что мы поступаем именно так при создании и назначении нового задания на обработку? Очевидно, что все зависит от расположения рабочих процессов. Если все рабочие процессы равномерно распределены между узлами (именованными экземплярами среды выполнения Erlang), причем нагрузка и среднее время выполнения задач на этих узлах примерно одинаковы, то в таком случае мы можем назначить новое задание на обработку освободившемуся процессу. При этом мы действительно получим выигрыш от использования многозадачности при обработке исходных данных, т. к. ситуация, при которой обработка некоторой порции одним из процессов тормозит весь процесс обработки исходных

данных, вряд ли возникнет. Если же конфигурация рабочих процессов такова, что выполнение одного и того же задания разными рабочими процессами сильно отличается, то подход, при котором мы новое задание назначаем освободившемуся процессу, неприменим (если мы не хотим получить вместо ускорения процесса обработки его замедление). Понятно, что в таком случае нам необходимо распределять задания между рабочими процессами каким-то другим способом. Иными словами, нам нужен некоторый механизм распределения заданий в зависимости от того или иного критерия.

Для начала рассмотрим, какие существуют механизмы для решения подобных проблем в других средах времени выполнения (и операционных системах). В большинстве таких сред создание новой единицы выполнения кода (потока либо процесса) является достаточно затратной операцией (по сравнению с процессами среды выполнения Erlang). Следует сказать, что затраты на создание новой единицы выполнения кода (потока либо процесса) различаются в зависимости от среды. Так, например, в операционной среде Linux затраты на создание нового процесса достаточно малы, поэтому в серверных приложениях для обработки запроса от клиента нередко создают новый процесс при помощи функции **fork(2)**. Более того, потоки в операционной системе Linux реализованы через процессы, разделяющие адресное пространство (и ряд других ресурсов) с родительским процессом. С другой стороны, если мы посмотрим на темную сторону силы (на операционные системы MS Windows), то в ней единицей выполнения кода является поток (а процесс является контейнером для потоков и ряда других ресурсов); причем создание потоков в операционных системах MS Windows является довольно дорогостоящей операцией. А для управляемых сред времени выполнения, таких как JVM, .NET Framework или Mono все более сложно: в них единицей выполнения кода является управляемый поток, который может соответствовать единицам выполнения кода операционной системы, а может и не соответствовать.

Но вне зависимости от среды выполнения (или операционной системы), мы не можем одновременно создать большое количество единиц выполнения кода, т. к. в худшем случае у нас закончатся системные ресурсы, а в лучшем — мы придем к ситуации, когда среда выполнения большую часть времени тратит на переключение контекста единиц выполнения кода. Поэтому в ситуации, когда необходимо создать единицу выполнения кода для выполнения некоторой задачи, рекомендуется использовать пул единиц выполнения кода. Обычно для планирования выполнения заданий используется пул потоков. В некоторых ситуациях можно использовать и пул процессов (если таковой есть); так, например, в языке Python для распараллеливания вычислений рекомендуется использовать пул процессов (воспользовавшись стандартным модулем *multiprocessing*) из-за GIL (Global Interpreter Lock, см. врезку).

А теперь вернемся к нашей проблеме: нам необходим некоторый механизм назначения заданий рабочим процессам в зависимости от того или иного критерия. В отличие от большинства других сред времени выполнения (и операционных систем), идеология среды выполнения Erlang подразумевает, что мы можем

Global Interpreter Lock

Выражение Global Interpreter Lock (GIL) переводится как «глобальная блокировка интерпретатора». Это средство синхронизации, используемое потоком интерпретатора некоторого языка программирования, чтобы избежать совместного выполнения небезопасного (с точки зрения многопоточности) кода другими потоками. Как

следует из названия, данная блокировка существует в единственном экземпляре в пределах процесса интерпретатора. Следствие ее применения — тот факт, что для параллельного выполнения некоторых задач надо использовать процессы, а не потоки. Примерами интерпретаторов языков с GIL являются CPython и CRuby.

МНОГО ЗАДАЧ

создавать столько процессов, сколько нужно (однако не следует забывать, что есть ограничение на количество процессов для каждого экземпляра среды выполнения Erlang), не опасаясь роста накладных расходов на поддержание их работы. Поэтому для нас нет смысла в пуле процессов (языка Erlang): нам проще создать новый процесс, чем иметь набор предопределенных процессов.

С другой стороны, если на одном узле (в одном экземпляре среды выполнения Erlang) мы будем создавать все новые и новые процессы, то начиная с какого-то момента останемся без прироста производительности (вызванного одновременным выполнением задач). Связано это с тем, что количество процессоров и/или ядер процессоров ограничено и обычно достаточно мало. Максимальное количество одновременно работающих единиц выполнения кода равно сумме всех ядер всех процессоров в системе (или произведению количества процессоров на количество ядер на одном процессоре, если у всех процессоров одинаковое количество ядер). И если количество единиц выполнения кода, запланированных для выполнения, превышает это максимальное количество, то некоторые из единиц выполнения кода будут простаивать, а система должна будет затрачивать время на переключение с одной единицы выполнения кода на другую, чтобы все единицы выполнения кода получили процессорное время.

Конечно, для среды времени выполнения Erlang временем переключения процессов можно пренебречь, но все остальное будет справедливо: чем больше процессов мы создадим, тем больше их будет простаивать в ожидании шанса на выполнение.

Очевидно, что в такой ситуации мы должны создавать процессы на разных узлах (именованных экземплярах среды выполнения Erlang), причем эти узлы должны располагаться на разных компьютерах, иначе все процессы будут выполняться на одних и тех же физических ядрах и процессорах, что будет эквивалентно ситуации с одним экземпляром среды выполнения. Поэтому наш механизм назначения заданий должен выбирать (или уметь создавать в определенных ситуациях) некоторый узел, на нем создавать процесс и выполнять с его помощью интересующее нас задание. Обычно для работы такого механизма назначения задач необходимо создать некоторый предопределенный набор узлов. Из этого набора узлов механизм назначения заданий будет по какому-либо критерию выбирать наиболее подходящий узел и выполнять задание на нем (по такому же принципу работают и другие пулы, например, пул потоков). И, соответственно, мы будем называть этот механизм пулом узлов.

В нашем очередном примере в качестве механизма назначения заданий мы и будем использовать пул узлов. Это означает, что наш пример будет абстрагирован от выбора процесса для выполнения очередного задания. Более того, при правильной реализации примера мы сможем легко поменять этот механизм назначения заданий на какой-либо другой, не затрагивая код примера. При реализации нашего примера мы поступим точно так же, как поступали раньше: создадим модуль (в нашем примере это будет модуль `parallel_pool_helper`), содержащий общую функциональность, на основе которой мы в дальнейшем реализуем многозадачные версии функций `map` и `reduce`.

Альтернатива для `rsh`

Для создания узлов на удаленных хостах среда времени выполнения Erlang использует утилиту `rsh`. Если данной утилиты у пользователя нет и/или есть желание использовать альтернативу этой про-

граммы, то это можно сделать следующим способом. При старте экземпляра среды выполнения Erlang следует добавить ключ `-rsh Program`, где `Program` — аналог утилиты `rsh`.

В качестве следующего шага, определим ряд записей, чтобы сделать код более понятным (причем эти записи определены только на уровне модуля `parallel_pool_helper`):

```
-record(tasks_descr, {created=0, processed=0, rest=[]}).
-record(task_result, {index, result}).
```

Запись `task_descr` предназначена для хранения данных о процессе обработки исходного списка: поле `created` содержит количество созданных заданий, поле `processed` — количество выполненных заданий, поле `rest` — остаток необработанных данных. Очевидно, что когда в поле `rest` пустой список, новых заданий создавать не надо; если поля `created` и `processed` содержат одно и то же значение — значит, мы обработали все исходные данные и результаты обработки сохранили в промежуточном хранилище. Запись `task_result` определена для передачи результатов обработки исходных данных: поле `index` содержит индекс порции данных, поле `result` — результат обработки порции исходных данных. Отметим, что, в отличие от предыдущего примера (из `LXF170`), запись `task_result` не содержит идентификатор рабочего процесса, обработавшего данную порцию данных. Дело в том, что там мы этому рабочему процессу назначали новую порцию данных; а сейчас мы перекладываем эту ответственность на внешний механизм назначения заданий (на пул узлов, в нашем случае).

Теперь давайте создадим функцию, которую будут выполнять рабочие процессы. Это будет экспортируемая функция `parallel_pool_helper:pool_worker/3`:

```
pool_worker(Fun, Portion, Index, Master) ->
  Result = Fun(Portion),
  Master ! #task_result{index=Index, result=Result}.
```

Она проще соответствующей функции для рабочих процессов из предыдущего примера: в ней нам нет необходимости организовывать цикл обработки сообщений от родительского процесса. Связано это с тем фактом, что в данном примере мы не управляем рабочими процессами напрямую, а полагаемся на некоторый механизм назначения заданий (например, на пул узлов). Это, в частности, означает, что мы не знаем ничего о рабочих процессах, которые обрабатывают наши порции данных (не знаем идентификаторы этих процессов). Как уже говорилось, эта функция экспортируется из модуля `parallel_pool_helper`, а в модуле `parallel_pool_helper` мы хотим полностью абстрагироваться от механизма назначения заданий; поэтому клиенты нашего модуля должны знать, какую функцию должны выполнять рабочие процессы при назначении конкретного механизма распределения заданий. Конечно, мы можем убрать эту зависимость клиентов от функции

»

» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)

Про файл `.hosts.erlang`

Файл `.hosts.erlang` содержит имена хостов, записанные в виде выражений языка Erlang. Формат его таков: каждая строка содержит только одно имя хоста, заключенное в одиночные кавычки, и заканчивается точкой (т.к. это выражение языка Erlang). Последняя строка файла должна быть пустой. Вот пример такого файла:

```
'host2.somecorp.ru'.
<пустая строка>
```

Файл `.hosts.erlang` может располагаться в следующих местах системы: в текущем рабочем каталоге, в домашнем каталоге Erlang/OTP (в `$OTP_ROOT`). Именно в таком порядке происходит поиск файла `.hosts.erlang`, пока он не будет найден.

для выполнения рабочими процессами и передавать ее через определенный нами интерфейс взаимодействия (этот интерфейс является просто функцией) с механизмом назначения заданий, но при этом код станет более сложным, хотя при создании реальных программ указанный подход предпочтительнее.

В качестве следующего шага создадим ряд функций для инкапсуляции таких операций, как создание и назначение новой задачи рабочему процессу и сохранение результата обработки порции в промежуточном хранилище. Это будут функции `collect_result/3` и `assign_task/4`, определенные в модуле `parallel_pool_helper`, но не экспортируемые из него:

```
collect_result(Result, Index, Storage) ->
array:set(Index, Result, Storage).
assign_task(Source, PortionSize, Index, AssignFun) when
length(Source) =< PortionSize ->
AssignFun(Source, Index, self()),
[];
assign_task(Source, PortionSize, Index, AssignFun) ->
{Portion, Rest} = lists:split(PortionSize, Source),
AssignFun(Portion, Index, self()),
Rest.
```

Функция `collect_result/3` сохраняет результат обработки порции в промежуточном хранилище (массиве). Эта функция ничем не отличается от подобной функции из предыдущего примера. Функция `assign_task/4` служит для создания и назначения задания на обработку порции данных рабочему процессу. На вход она принимает необработанный остаток исходных данных `Source`, размер порции `PortionSize`, индекс очередной порции данных `Index` и интерфейс (интерфейсную функцию) к механизму назначения заданий `AssignFun`. В функции `assign_task/4` мы выделяем очередную порцию данных, отдаем ее в виде задания на обработку рабочему процессу и возвращаем остаток исходных данных после создания задания. Понятно, что при этом в функции `assign_task/4` мы должны обрабатывать два возможных случая: когда размер остатка исходных данных больше размера порции данных и когда размер остатка исходных данных меньше или равен размеру порции данных. В первом случае мы разбиваем остаток исходных данных на порцию, которую будет обрабатывать рабочий процесс, и новый остаток; во втором случае весь остаток будет обработан, а новым остатком будет пустой список. Для назначения задания рабочему процессу мы используем интерфейс к механизму назначения заданий, который является функцией `AssignFun`. В нее мы передаем очередную порцию для обработки, индекс этой порции и идентификатор главного процесса. В этом заключается отличие функции `assign_task/4` от подобной функции, определенной в предыдущем примере.

В нашем примере мы используем следующий подход к назначению заданий рабочим процессам: как только мы получаем

результат обработки какой-либо порции, мы назначаем новое задание на обработку очередной порции. При этом количество рабочих процессов, обрабатывающих порции исходных данных, ограничено (и гораздо меньше количества этих порций). Но чтобы этот подход работал, необходимо одно предусловие: мы должны создать и назначить рабочим процессам определенное количество (равное количеству одновременно выполняющихся рабочих процессов) заданий на обработку. Для этого мы создадим функцию `distribute_init_tasks/4` в модуле `parallel_pool_helper` (но не экспортируем ее из этого модуля):

```
distribute_init_tasks(#tasks_descr(created=Created, rest=[]), _
PortionSize, _AssignFun, _WorkerCount) ->
#tasks_descr(created=Created, rest=[]);
distribute_init_tasks(#tasks_descr(created=WorkerCount,
rest=Rest), _PortionSize, _AssignFun, WorkerCount) ->
#tasks_descr(created=WorkerCount, rest=Rest);
distribute_init_tasks(#tasks_descr(created=Created, rest=Source),
PortionSize, AssignFun, WorkerCount) ->
Rest = assign_task(Source, PortionSize, Created, AssignFun),
TasksDescr = #tasks_descr(created=Created+1, rest=Rest),
distribute_init_tasks(TasksDescr, PortionSize, AssignFun,
WorkerCount).
```

При работе функции `distribute_init_tasks/4` возможно возникновение следующих 3-х ситуаций, которые эта функция должна уметь обрабатывать (именно поэтому она содержит 3 варианта). Во-первых, у нас остаток необработанных исходных данных может быть пустым. Во-вторых, мы можем создать необходимое количество порций данных и раздать их рабочим процессам в качестве заданий на обработку. И, наконец, возможна ситуация, когда у нас есть еще исходные данные для обработки, и мы не создали необходимого количества заданий для обработки.

Очевидно, что в первых двух ситуациях следует завершить выполнение функции `distribute_init_tasks/4`; в этом случае мы возвращаем соответствующие данные о процессе обработки исходного списка (экземпляр записи `task_descr`). В последнем случае, мы создаем очередную порцию, назначаем ее в качестве задания какому-то рабочему процессу и рекурсивно вызываем сами себя. Так как при каждом вызове третьего варианта функции `distribute_init_tasks/4` у нас объем необработанных исходных данных уменьшается (на размер порции данных), а количество созданных задач на обработку увеличивается на 1, то, в конце концов, мы придем к вызову либо первого варианта, либо второго варианта функции `distribute_init_tasks/4`. Это означает, что функция `distribute_init_tasks/4` написана корректно и не приведет к бесконечному рекурсивному вызову самой себя.

После того, как мы научились инициализировать рабочие процессы необходимым количеством заданий, пришла пора организовать взаимодействие между рабочими процессами и главным. Это взаимодействие заключается в следующем: мы должны ожидать сообщения от рабочих процессов с результатами обработки порций данных. При получении такого сообщения мы должны сохранять результаты обработки порций в промежуточное хранилище, после чего, если еще есть необработанные данные, создавать новое задание и отправлять его на обработку. Обрабатывать это задание будет не обязательно тот процесс, от которого пришло сообщение с результатом обработки некоторой предыдущей порции. Соответственно, у нас возможны следующие 3 ситуации: у нас есть еще исходные данные для обработки; исходные данные для обработки закончились, но не все рабочие процессы закончили выполнение своих заданий (обработку порций исходных данных); и, наконец, у нас закончились все исходные данные и все рабочие процессы завершили свою работу. Очевидно, что

» Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.

в первом случае мы будем создавать новые задания на обработку исходных данных; во втором случае мы этого делать не будем; а в третьем случае — закончим работу (и вернем хранилище промежуточных данных, которое будет содержать результаты обработки всех порций данных). Это взаимодействие между рабочими процессами и главным процессом с описанным выше поведением мы реализуем в функции `handle_workers/4`, определенной в модуле `parallel_pool_helper`, но не экспортируемой из него:

```
handle_workers(#tasks_descr{created=N, processed=N, rest=[]},
Storage, _PortionSize, _AssignFun) ->
Storage;
handle_workers(#tasks_descr{created=C, processed=P, rest=[]},
Storage, PortionSize, AssignFun) ->
receive
#task_result{index=Index, result=Dest} ->
UpdatedStorage = collect_result(Dest, Index, Storage),
TasksDescr = #tasks_descr{created=C, processed=P+1,
rest=[]},
handle_workers(TasksDescr, UpdatedStorage, PortionSize,
AssignFun);
_ -> handle_workers(#tasks_descr{created=C, processed=P,
rest=[]}, Storage, PortionSize, AssignFun)
end;
handle_workers(#tasks_descr{created=C, processed=P, rest=Src},
Storage, PortionSize, AssignFun) ->
receive
#task_result{index=Index, result=Dest} ->
UpdatedStorage = collect_result(Dest, Index, Storage),
Rest = assign_task(Src, PortionSize, C, AssignFun),
TasksDescr = #tasks_descr{created=C+1, processed=P+1,
rest=Rest},
handle_workers(TasksDescr, UpdatedStorage, PortionSize,
AssignFun);
_ -> handle_workers(#tasks_descr{created=C, processed=P,
rest=Src}, Storage, PortionSize, AssignFun)
end.
```

Видно, что функция `handle_workers/4` содержит 3 варианта, каждый из которых обрабатывает одну из 3-х возможных ситуаций во время взаимодействия между главным процессом и рабочими процессами.

А теперь пришла пора объединить все вместе: давайте создадим точку входа для общей функциональности, на основе которой создадим очередные версии функций `map` и `reduce`. Это будет функция `parallel_pool_helper:pool_core/5`, определенная в модуле `parallel_pool_helper` и, что естественно, экспортируемая из него:

```
pool_core(FinalAggrFun, Source, PortionSize, AssignFun,
WorkerCount) ->
process_flag(trap_exit, true),
PortionCount = parallel_common:calc_portion_
count(length(Source), PortionSize),
TasksDescr = distribute_init_tasks(#tasks_descr{rest=Source},
PortionSize, AssignFun, WorkerCount),
EmptyStorage = array:new([size, PortionCount], {fixed, true},
{default, none}),
FullStorage = handle_workers(TasksDescr, EmptyStorage,
PortionSize, AssignFun),
process_flag(trap_exit, false),
FinalAggrFun(array:to_list(FullStorage)).
```

В этой функции мы создаем и назначаем рабочим процессам определенное количество заданий (равное значению параметра `WorkerCount`), после чего взаимодействуем с рабочими процессами, пока все исходные данные не будут обработаны рабочими процессами (и все результаты обработки не будут собраны в промежуточном хранилище); и, наконец, из результатов обработки порций формируем итоговый результат. В качестве одного из параметров,

в функцию `parallel_pool_helper:pool_core/5` мы передаем интерфейс (а точнее, интерфейсную функцию) к механизму назначения заданий. Этим параметром является параметр `AssignFun`. Это означает, что определенная нами общая функциональность не зависит от используемого механизма назначения заданий, и любой клиент, использующий эту нашу общую функциональность, может использовать любой удобный ему механизм. После создания необходимой общей функциональности, определенной в модуле `parallel_pool_helper`, мы можем перейти к созданию на основе этой функциональности очередных версий функций `map` и `reduce`. Но прежде чем их создавать, следует выбрать механизм назначения заданий. Как уже говорилось выше, логично выбрать некоторый пул узлов. В стандартной библиотеке языка Erlang такой пул узлов есть: он определен в модуле `pool`. Этот пул узлов может как самостоятельно создавать узлы на определенных хостах, так и использовать некоторые уже созданные узлы.

Для создания узлов на хостах используется программа `rsh` (или ее альтернатива). Поэтому, если у клиента (на каких-либо узлах) не установлена эта программа (или не указано использование ее альтернативы) или если клиент находится на темной стороне силы (использует операционную систему семейства Microsoft Windows), то пул узлов создавать узлы на этих хостах не сможет. В этом случае клиент может сам создать необходимое количество узлов на этих хостах и приказать пулу узлов использовать их.

Для начала работы с пулом узлов необходимо вызвать одну из функций `pool:start/1` или `pool:start/2`. Если есть такая возможность (см. врезку про программу `rsh` на стр. 85), на каждом хосте, описанном в файле `.hosts.erlang`, будет создан свой узел. Если при старте пул узлов создать узлы не смог или же есть необходимость добавить дополнительные узлы в пул, можно воспользоваться функцией `pool:attach/1`, которая добавляет в пул узлов узел, переданный в качестве аргумента.

Для выполнения некоторого задания (вычисления функции, определенной в некотором модуле, с заданными аргументами) используется одна из функций `pool:pspawn/1` или `pool:pspawn_link/3`. Обе эти функции создают процесс на одном из узлов пула, где ожидается наименьшая загрузка, и возвращают идентификатор созданного процесса, что позволяет дополнительно с ним взаимодействовать. Их отличие только в том, что первая функция просто создает процесс, а вторая, помимо процесса, создает еще и связь между созданным и вызывающим процессом. И, наконец, чтобы закончить работу с пулом узлов, необходимо вызвать функцию `pool:stop/0`; при этом все узлы пула будут уничтожены.

В очередной раз мы вынуждены остановиться из-за того, что место, выделяемое для статьи, ограничено. В следующий раз мы закончим пример на основе пула узлов, который начали в этом номере, а также подведем итоги решения задачи по созданию многозадачных версий функций `map` и `reduce`. **LXF**

Альтернативная инициализация

Пример, приведенный в данной статье, по своей структуре похож на пример из **LXF170**. Как и там, задания на обработку создаются у нас в двух местах: во-первых, в функции `distribute_init_tasks/4` для создания необходимого количества рабочих процессов (для их инициализации); во-вторых, в функции `handle_workers/4` для организации взаимодействия между рабочими процессами и главным процессом (и для обработки оставшихся данных, естественно). Если кого-то не устраивает,

что создание заданий на обработку у нас «размазано» по двум функциям, можно создать альтернативный вариант инициализации рабочих процессов: инициализировать их некоторыми «псевдозаданиями». Пример такой альтернативной инициализации мы приводили в **LXF170**. Похожую альтернативную инициализацию рабочих процессов можно сделать и для текущего примера, но мы оставим такую реализацию в качестве домашнего задания для тех, кому это интересно.



Кластеры: Легко

Динамический и виртуальный кластеры



Наш эксперт

Андрей Сухарев
Физик, участник экспериментов CO PAH и CERN. Любит свою работу за переплетение с увлечениями — программирование и администрирование Linux-систем.

Были бы данные, а **Андрей Сухарев** и **Евгений Балдин** кластер для их параллельной обработки соорудить сумеют.

У вас есть данные, которые можно обрабатывать параллельно и независимо? Есть поблизости дружелюбный кластер? Тогда можно еще сильнее упростить себе жизнь и размножить привычное программное окружение сразу на нем.

Введение

Одним из важных классов параллельных вычислений является так называемый портфель задач. Каждая задача в этом случае является независимой единицей, а распараллеливание производится на уровне данных. Это возможно, если данные являются независимыми. Задачи помещаются в портфель и достаются оттуда, как только освобождается необходимый для ее исполнения ресурс. Такие вычисления замечательно масштабируются — надо только как-то суметь захватить доступные компьютерные ресурсы.

В современном мире фактически каждый физик/химик/биолог и даже экономист (если он ученый) должен быть немного информатиком: бурно развивающиеся вычислительные ресурсы расширяют возможности как экспериментатора, так и теоретика. Безусловно не заменяют, но являются весьма ценным дополнением к уже имеющимся инструментам.

Одной из особенностей долго идущих экспериментов является разнородность компьютерной инфраструктуры и постепенно разрастающиеся связи. Пример схемы такого окружения можно увидеть на рис. 1. Там интересны три кружочка в левом нижнем углу — это три независимых суперкомпьютерных кластера, которые можно использовать для своих нужд. И здесь вылезает основная проблема: ПО, как и версии GNU/Linux, на этих вычислительных ресурсах различны. Нет ни времени, ни желания выяснять, а нет ли каких-либо багов, специфичных только для этой коллекции библиотек, поэтому проще принести свое привычное окружение, с уже известными и полюбившимися багами, прямо туда. Также следует учитывать, что кластеры на стороне не являются собственностью исследователей, поэтому как только необходимость в вычислениях отпадает, то ресурсы нужно автоматически освободить. Особенно это важно в случаях, если требуется высокая пиковая производительность, но лишь изредка.

Разберем одну из реализаций того, как можно задействовать эти вычислительные мощности в условиях ограниченных денежных и человеческих ресурсов. Она не обязательно будет оптимальной для вашей задачи, и наверняка есть решения получше, но была успешно собрана, исправно работает и развивается.

Динамический кластер

Естественным решением является виртуализация. Надо только развернуть виртуализованную среду, научиться запускать ее на внешнем вычислительном кластере как задачу тамашней батч-системы и, наконец, автоматизировать весь цикл «пользовательское задание → запуск VM → работа VM → освобождение ресурсов». Чтобы получить оптимальную конфигурацию виртуализованной среды, способной обеспечить и высокую эф-



► Рис. 1. Структура суперкомпьютерной сети ННЦ. Зелеными линиями показаны основные 10 GbE соединения, а пунктиром — планируемые группы и связи с ними. Отмечены точки сбора статистики.

фективность использования ЦПУ физических узлов, и стабильную работу виртуальных машин (VM), мы изучили три системы виртуализации: *VMware Server* (<http://www.vmware.com/products/server/>), *Xen* (<http://xenproject.org/>) и *KVM* (<http://www.linux-kvm.org/>). Для постоянного использования мы выбрали *KVM*. Главным и определяющим ее преимуществом стала простота применения: *KVM*, как правило, уже входит в состав современных дистрибутивов GNU/Linux, а значит, проста в установке. В отличие от *Xen*, *KVM* не требует специального модифицированного ядра GNU/Linux, установка которого на кластер бывает и невозможна — например, из-за конфликта с драйверами InfiniBand. *KVM*-решение прошло широкомасштабное тестирование, в котором до 512 двудерных VM одновременно выполняли до 1024 заданий обработки данных и моделирования нашего домашнего эксперимента КЕДР.

Общий вид схемы интеграции систем управления пакетной обработкой заданий показан на рис. 2. Специальный сервис периодически проверяет наличие ожидающих выполнения заданий в очереди батч-системы эксперимента. При появлении таких заданий он ставит в очередь системы пакетной обработки кластера соответствующее количество так называемых «заданий запуска VM». Каждое «задание запуска VM» полностью занимает один физический узел кластера и запускает несколько VM, по количеству физических ядер ЦПУ. VM загружаются, сообщают о своей готовности системе пакетной обработки эксперимента, и начинают выполняться задания пользователей. Когда заданий в очереди не остается, инициируется процедура самовыключения VM, и ресурсы кластера освобождаются для других пользователей.

Пара слов о KVM

Kernel-based Virtual Machine или *KVM* — это программное решение, обеспечивающее виртуализацию в среде GNU/Linux на платформе x86, которая поддерживает аппаратную виртуализацию



Наш эксперт

Андрей Грозин
Доктор физ.-мат. наук. Стал разработчиком Gentoo, сдав 2 письменных экзамена и устный, причем устный — только со второй попытки (последний раз его выгоняли с экзамена лет за 30 до того).



Наш эксперт

Евгений Балдин
Физик, который действительно знает, что такое нехватка вычислительных ресурсов.

СДЕЛАТЬ САМИМ

на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine). Очень полезной опцией *KVM* является использование дисковых образов VM в режиме *snapshot*. При этом базовый образ во время работы VM остается неизменным, а все изменения записываются во временный файл; причем если этот файл не указать явно, то он создается в `$TMPDIR` и автоматически исчезнет после остановки VM. Это дает сразу две выгоды: во-первых, разместив один образ диска на системе хранения вашего дружественного кластера, можно запускать столько VM, сколько вам разрешат; а во-вторых, если внутри VM что-то сломалось, то все это ликвидируется после простого перезапуска.

Готовим виртуальную машину

Создание рабочего дискового образа VM не представляет сложности и хорошо документировано. В рамках нашего эксперимента было отдано предпочтение образам формата *raw*, к содержанию которых довольно легко получить доступ, не запуская VM (обратитесь к **man lmount**). Пустой файл для такого образа можно создать командой `dd if=/dev/zero...` GNU/Linux устанавливается стандартно, путем загрузки VM с установочного образа. Для этой процедуры можно воспользоваться каким-либо вспомогательным ПО — например, *virt-manager* (<http://virt-manager.org/>).

Первое, о чем придется позаботиться в условиях динамического кластера — настройка сети. Виртуальные машины под управлением *KVM* могут иметь сетевые устройства, обслуживаемые различными драйверами — например, *e1000* или паравиртуальный *virtio*. В зависимости от версии *KVM* на физическом вычислительном узле и от версии ядра GNU/Linux в виртуальной машине предпочтительнее могут быть разные варианты. Чтобы оставить возможность легко менять драйвер, поступим так:

```
device=>eth0>
mac=/sbin/ifconfig $device | /bin/awk '/HWaddr/ { print $5 }'
if [ -z "$mac" ]; then
    echo "no $device, try with driver e1000..."
/sbin/modprobe e1000
mac=/sbin/ifconfig $device | /bin/awk '/HWaddr/ { print $5 }'
fi
```

Поскольку дисковый образ у всех VM один, единственное, чем они могут отличаться друг от друга, это индивидуальный MAC-адрес, задающийся в командной строке запуска каждой VM. Сетевые настройки VM полностью выводятся из него — какой будет у VM IP-адрес, какой сетевой шлюз, какие маршруты.

Для каждого физического узла кластера выделяется поддиапазон IP-адресов VM. Физический узел работает маршрутизатором между VM и внешней сетью. Внешняя же сеть может быть какой угодно. Когда внутри VM появился доступ к сети, все необходимое стандартное ПО можно установить из репозитория дистрибутива — ровно так же, как и в случае реальной машины.

В случае каких-то сбоев при старте может оказаться так, что *KVM*-процесс запустился и выполняется, а сама VM недоступна. Чтобы не занимать ресурсы зря, такие VM надо сразу останавливать. Для этого применяются:

- » параметр ядра `panic=15` (перезагрузка VM в случае паники ядра);
- » параметр командной строки *KVM* `-no-reboot` (выключение VM при перезагрузке);
- » *сгол*-задача периодической проверки, работает ли на VM демон системы управления заданиями. Если это не так, VM выключается.

В остальном настройка виртуальной машины ничем не отличается от обычной. Если специальное ПО находится на NFS-серверах, надо настроить доступ к ним. Требуется внести адреса и имена VM в DNS, зарегистрировать их в батч-системе в качестве вычислительных узлов, убедиться, что ходит почта, и т. п.

Запуск VM

Когда задание запуска VM распределено батч-системой на конкретный физический узел кластера, начинается работа скрипта запуска VM. Он выполняет следующие действия:

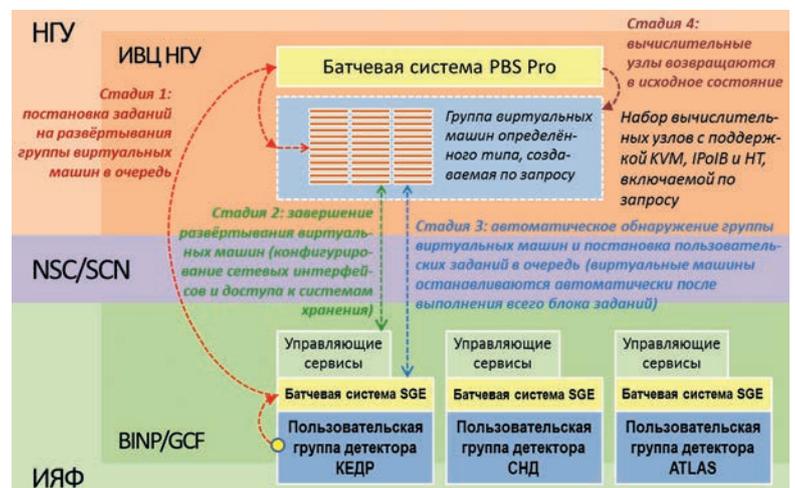
- 1 Определяет количество процессоров и памяти на узле, куда он попал.
- 2 Решает, сколько VM он может запустить в этих условиях. Виртуальные машины тоже могут быть многоядерными, например, вы запускаете две 4-ядерных VM на 8-ядерном узле. Память делится между VM поровну, и примерно 20 % оставляется для работы собственной системы физического узла.
- 3 Запускает виртуальные машины.

Поскольку задание работает от имени обычного пользователя, а запуск VM требует более высоких привилегий, используется механизм **sudo**. Затем скрипт ждет завершения работы VM или истечения времени, отведенного на задание.

Команда запуска отдельной VM внутри батч-задания выглядит примерно так:

```
/usr/libexec/qemu-kvm -cpu qemu64 -name $name-$vncnbr -vnc :$vncnbr -daemonize -drive file=$file,if=ide,index=0,boot=on, snapshot=on -m $mem -smp $smp -no-reboot -monitor telnet:127.0.0.1:$monitor,server,nowait -net nic,macaddr=$mac, model=$model -net tap,script=$script,downscript=$downscript, vlan=0,ifname=$itf drive file=$file2,if=ide,index=1,boot=off, snapshot=on
```

Как видно, для доступа к консоли VM может использоваться VNC. Мониторный порт, работающий в режиме telnet, используется заданием запуска виртуальных машин для отправки им различных сигналов. В данном примере для дисковых образов используется интерфейс *ide*, но это может быть и *virtio* — в зависимости от того, что позволяют вам ваша версия *KVM* и ваша версия Linux в VM.



» Рис. 2. Схема интеграции систем управления заданиями на группы ИЯФ и кластера ИВЦНУ. Показаны четыре стадии цикла развертывания VM на стороне ИВЦНУ.

Скрипт создания сетевого устройства:

```
#!/bin/sh
if [ -n "$1" ]; then
  /usr/sbin/tunctl -u root -t $1 2>&1 | grep -v 'TUNSETIFF: Device
or resource busy'
  /sbin/ifconfig $1 0.0.0.0 up
  sleep 0.5s
  /sbin/ifconfig $1
  exit 0
else
  echo "Error: no interface specified"
  exit 1
fi
```

После запуска VM завершается настройка сети:

```
# маршрут к VM
/sbin/route add -net 172.16.${block_id}.${vm_id}netmask
255.255.255.255 dev ${itf}
# другие нужные маршруты
...
# не забываем про ip-forwarding
/sbin/sysctl -w net.ipv4.conf.${external_interface}.forwarding=1
/sbin/sysctl -w net.ipv4.conf.${itf}.forwarding=1
```

Сервис-интегратор

С локальной батч-системой сервис-интегратор взаимодействует через интерфейс командной строки — запускает команды и анализирует их вывод. Он в цикле периодически проверяет наличие новых пользовательских заданий в очереди; определяет, сколько VM сейчас запущено и на каких физических узлах; решает, нужны ли новые VM. Взаимодействие с внешними батч-системами тоже осуществляется через интерфейс командной строки, пропущенный через ssh-канал. Сервис не запоминает своего состояния, начинает каждый новый цикл с чистого листа и может быть спокойно остановлен и вновь запущен в любое время.

Использование командной строки для взаимодействия с локальной батч-системой — один из недостатков данного решения. Оно приводит к повышенной нагрузке на ее сервер управления и чревато сбоями. Идеальным вариантом было бы встроить ин-

тегратор непосредственно в батч-систему, но это требует детального разбирательства того, как она устроена.

Остановка VM

Остановка VM должна происходить автоматически, если:

- » отсутствуют новые пользовательские задачи, подходящие для выполнения на данной VM;
- » истекло время, выданное батч-системой кластера;
- » произошел сбой VM.

Одно задание запуска VM занимает целиком один многоядерный физический вычислительный узел кластера. Как правило, в рамках задания запускается несколько VM. После этого запускающий скрипт продолжает свою работу, периодически проверяя, не завершились ли VM и не пора ли завершать их принудительно, если время, отпущенное на задание, истекает.

Выключение VM при отсутствии подходящих заданий в локальной батч-системе производится через саму эту систему. Сервис-интегратор запускает задание, выключающее блок VM, работающих на одном физическом узле. Если окажется, что за время принятия решения об остановке этих VM батч-система уже успела распределить на какую-то из них настоящее задание, задание выключения будет просто отброшено.

Если скрипт запуска VM прождал их завершения до конца отведенного на задание времени, он посылает всем VM через мониторинговый интерфейс сигнал **system_powerdown**, а затем, если VM не смогли по какой-то причине выключиться нормально — сигнал **quit**, вызывающий немедленную остановку *KVM*. (Заметим, что, поскольку дисковые образы используются в режиме snapshot, это не приведет к повреждениям файловой системы.) В конце концов, когда все VM на узле остановлены, скрипт запуска завершает свою работу, а с ним завершается и задание. Физический узел свободен для новых заданий.

Итог

В целом система вышла довольно надежной, хотя и требующей присмотра. Она легко расширяема — ограничивающим фактором здесь будут скорее возможности хранилища данных. Усовершенствования вносятся в процессе работы по мере накопления опыта.

Кластер на коленке

Если по соседству кластеров нет, Андрей Грозин обеспечивает распределенные вычисления своими руками.

У вас нет под рукой готового и хорошо настроенного кластера? А читать хочется, и вы имеете представление о Python? Тогда можно собрать кластер «на коленке».

Почему именно Python? К достоинствам этого скриптового высокоуровневого активно развиваемого языка программирования можно отнести то, что первые полезные результаты появятся минут через 5–10 после начала работы. Очень многие вещи, в том числе и касающиеся высокоэффективных вычислений, не надо сочинять самому, потому что они есть в стандартных и не очень стандартных библиотеках «питона».

Есть и несколько библиотек для организации «многомашинных» вычислений. Этот текст — краткий обзор возможностей пакета RPyC (Remote Python Call, <http://rpyc.sourceforge.net/>). Это очень простой способ организации «кластера», а так как он может работать везде, где есть Python, то в свою вычислительную сеть можно объединить почти все, что обладает процессором, даже хоть сколько-нибудь продвинутые телефоны и домашние маршрутизаторы.

В классическом способе применения RPyC на всех облюбованных компьютерах запускаются ведомые [slave] серверы, а клиент обращается к ним и поручает выполнить некую работу. Ведомым серверам дозволено то же, что и интерпретатору Python на этой машине, без ограничений.

Важное предупреждение! Все эксперименты с «наколенным кластером» следует запускать только в изолированной сети, где нет посторонних! Ведомый сервер (по поручению клиента) может делать все, что разрешено запустившему его пользователю — скажем, удалить все файлы этого пользователя. Никакой аутентификации клиента не производится. Это обратная сторона простоты. Для более стационарных расчетов присмотритесь к готовым системам управления распределенных вычислений.

Теперь начнем. Пусть в изолированной за межсетевым экраном домашней сети есть машины bilbo (основная) и gandalf (ведомая). Входим на gandalf через ssh и запускаем ведомый сервер:

```
gandalf> rpyc_classic
INFO:SLAVE/18812:server started on [0.0.0.0]:18812
```

Сервер ждет соединения от клиентов на порте 18812. Запускаем интерпретатор Python на bilbo и выполняем команды:

```
bilbo> python
Python 3.2.3 (default, Jan 20 2013, 20:05:19)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import rpyc
>>> gandalf=rpyc.classic.connect('gandalf')
```

После этого можно выполнять различные действия с bilbo на gandalf:

```
>>> gandalf.execute('n=2')
>>> gandalf.eval('n+1')
3
```

Можно использовать встроенные функции Python на gandalf, хотя интерпретатор работает на bilbo:

```
>>> gandalf_file=gandalf.builtins.open('/home/grozin/rpyc/
... 'mymodule.py')
```

Здесь на удаленной машине (gandalf) создан файл объект, а на локальной машине (bilbo) создана «сетевая ссылка» (прокси-объект) `gandalf_file` на него. Любые исполняемые над этой ссылкой действия переадресуются объекту на удаленной машине.

```
>>> print(gandalf_file.read())
#!/usr/bin/env python
from time import sleep
class MyClass:
    def __init__(self,t):
        self.t=t
    def f(self,n):
        sleep(self.t)
        return n+1
>>> gandalf_file.close()
```

Этот «прокси-объект» можно подставить в любую программу, ожидающую иметь файл объект. По фундаментальному для Python принципу утиной типизации «если объект ходит, как утка, плавает, как утка, и крикает, как утка, значит, он утка», этот объект — файл.

На `gandalf` можно использовать функции и прочие объекты из библиотечных модулей Python:

```
>>> gandalf_path=gandalf.modules.sys.path
>>> print(gandalf_path)
['/usr/bin', '/usr/lib/portage/pym', '/usr/lib64/python32.zip',
'/usr/lib64/python3.2', '/usr/lib64/python3.2/plat-linux2',
'/usr/lib64/portage/pym']
>>> gandalf_path.append('/home/grozin/rpyc')
```

Теперь `gandalf_path` — это прокси-объект для `sys.path` на `gandalf`, и любые изменения этого объекта сразу передаются на удаленную машину. Расширив `path`, можно использовать файл из этой директории на `gandalf`:

```
>>> gandalf_object=gandalf.modules.mymodule.MyClass(0)
>>> gandalf_object.f(3)
4
```

`gandalf_object` — это прокси-объект (сетевая ссылка) для объекта класса `MyClass` на машине `gandalf`. Его метод `f` прибавляет 1 к аргументу; чтобы мы могли моделировать длительные вычисления, он это делает за `t` секунд, где `t` — атрибут данного объекта.

```
>>> gandalf_object.t=2
>>> gandalf_object.f(4)
5
```

На сей раз нам пришлось ждать 2 секунды.

Через параметры можно передавать удаленным функциям любые объекты, в частности, локальные функции. Определим такую:

```
>>> def loc(n):
...     print('loc',n)
...     return n+1
```

Тогда функция `map` на `gandalf` на каждом шаге вызывает функцию `loc` на `bilbo` (callback):

```
>>> list(gandalf.builtins.map(loc,[1,2,3]))
loc 1
loc 2
loc 3
[2, 3, 4]
```

Все, что мы до сих пор обсуждали, несомненно, красиво — разные объекты могут жить на разных машинах, и единая программа работает с ними, не замечая этого. Но все эти операции синхронные: одна машина просит другую что-то сделать и ждет, когда та вернет ей результат. Для организации распределенных вычислений нужны асинхронные операции:

```
>>> gandalf_object.t=10
>>> async_f=rpyc.async(gandalf_object.f)
>>> res=async_f(1)
>>> res.ready
False
>>> # 10 секунд спустя
>>> res.ready
True
>>> res.value
2
```

Это уже лучше. Клиент может время от времени спрашивать, готов ли результат, и когда он будет готов, забрать его. Если запросить `res.value`, когда результат еще не готов, то клиент блокируется до момента, когда он будет готов:

```
>>> res=async_f(2)
>>> res.value
3
```

(после `res.value` 10 секунд ожидания, потом появляется ответ).

Но еще лучше определить callback-функцию, которая будет вызвана на локальной машине, когда результат будет готов:

```
>>> def callback(res):
...     print(res.value)
```

Эту функцию можно вызвать только в отдельном потоке [thread]:

```
>>> bgsrv = rpyc.BgServingThread(gandalf)
>>> res=async_f(3)
>>> res.add_callback(callback)
>>> # продолжаем что-то делать
4
>>> # продолжаем что-то делать
```

Это печать из функции `callback` из другого потока. Теперь этот поток можно и остановить:

```
>>> bgsrv.stop()
```

Например, на клиентской машине может работать графический пользовательский интерфейс (на Python такой написать легко, причем он будет работать на любой платформе — Linux, Windows, Mac — без малейших изменений в программе). Эта клиентская программа обращается к нескольким мощным серверам для проведения длительных вычислений, и регистрирует callback-функции, которые, например, добавляют очередную точку на график.

Наконец, закроем связь с машиной `gandalf`:

```
>>> gandalf.close()
>>>
bilbo>
```

Собственно говоря, вот и все. **1.25**

Обратная связь

Приглашаем высказаться потенциальных авторов статей по параллельным вычислениям — ценные предложения, критику и советы присылайте по электронной почте: E.M.Baldin@inp.nsk.su, A.G.Grozin@inp.nsk.su, A.M.Suharev@inp.nsk.su.

GIMP: Обработка

Александр Толстой решил, что его родословное древо следует украсить портретами высшего качества. *GIMP* ему в помощь!



Наш эксперт

Александр Толстой редактировал и переводил *LXF* на первых порах его существования, а теперь посвящает свободное время цифровой фотографии. И радуется за употребление на печати буквы «ё».



Старые фотографии по-прежнему пылятся в коробках из под обуви, но их все чаще хочется рассматривать на экране цифрового устройства. И нежеле в XXI веке нам отвлекаться на отсканированные вместе с фото грязь, пыль и сеточку рельефной бумаги. Избавимся же от этих дефектов с помощью *GIMP* и грамотного подхода.

Вместо предисловия

Руководство по исправлению поврежденных фотографий в Интернете не честь; поэтому, чтобы сказать здесь что-то новое, нужно как следует постараться. На данном уроке будет идти речь о работе в *GIMP* — самом мощном и популярном открытом графическом редакторе (он уже многие годы не сдается в борьбе за лавры *Photoshop*), при этом будут описаны интересные и неочевидные методы фоторетуши. Мы оцифруем и постараемся привести в пристойный вид черно-белые фотографии — уверен, что в вашем семейном архиве они имеются. В процессе обработки фото будет показано, как побороть часто встречающиеся дефекты, вызванные физическим старением снимка и особенностями его перевода в цифровой вид. Итак, поехали!

Убираем пыль и царапины

Главный враг фотолюбителя — это пыль, которая любит садиться как на сам отпечаток, так и на стекло вашего домашнего сканера.



Рис. 1. Фрагмент фотографии с дефектами: пыль отчетливо видна.

архивных фото

Знаете ли вы, что...

Модуль Refocus в свое время был интегрирован в программу *Digikam* — менеджер и каталогизатор фотографий для KDE. Если вы планируете обойтись без *GIMP*, то обратите внимание на *Digikam*. Refocus там спрятался во встроенном редакторе изображений (меню Улучшить > Резкость).

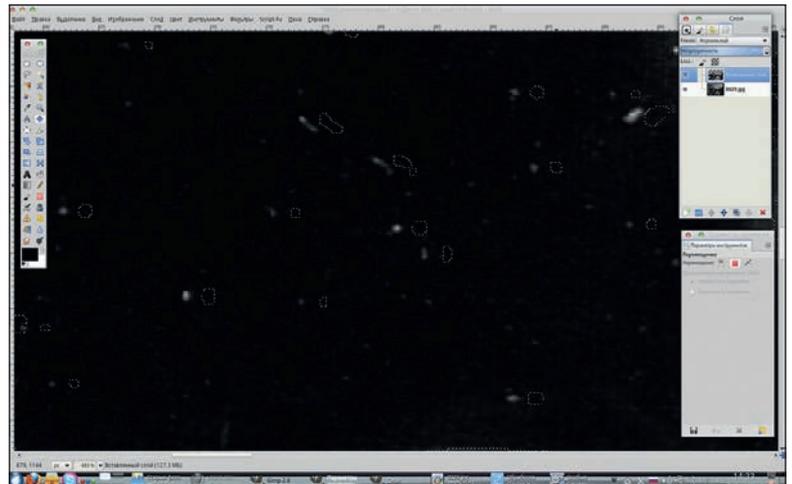
Поэтому, чтобы в дальнейшем пришлось меньше работать инструментом Штамп [Cloning brush], есть смысл аккуратно протереть неворсистой слегка влажной тканью как сам снимок, так и стекло сканера.

Итак, снимок отсканирован, и теперь у нас в *GIMP* он открыт во всей своей красе. Почти всегда на изображении будет какое-то количество царапин, отслоившихся участков, заломов и пылинок, поэтому классический совет в таких случаях — это аккуратно «замазать» все дефекты Штапмом, предварительно подобрав кисть нужного размера и жесткости. Однако есть и более эффективный метод, позволяющий достичь того же результата в полуавтоматическом режиме.

Рассмотрим пример на рис. 1, где присутствует хорошо различимая пыль.

Суть нашего метода такова: делается выделение всех дефектов по цвету, затем выделение смещается в сторону, откуда копируются участки «здорового» изображения. Затем скопированные участки перемещаются на место дефектов, загораживая их. Чтобы зритель не смог заметить границы таких участков, выделение с самого начала растушевывается. Такой метод хорошо работает на тех участках изображения, где яркость более-менее однородна. На практике есть смысл делать такую процедуру отдельно для темной части изображения и отдельно — для светлой. Рассмотрим более подробно, как это делается.

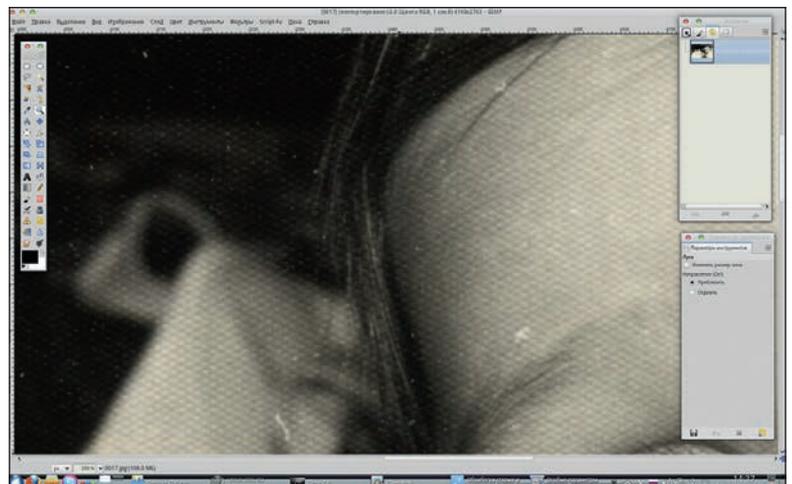
Выделим пыль и царапины. В *GIMP* есть инструмент Выделение по цвету (кнопка ) — с его помощью можно выделить все светлые участки на фото, а это нам и нужно, потому что пыль и царапины имеют белый или светло-серый цвет. Нам потребуется перевернуть инструмент в режим Добавить в текущее выделение (кнопка  в параметрах инструмента), после чего можно будет последовательно выбрать все цвета, которые относятся к пыли и царапинам. Чтобы получить более точный результат, есть смысл для выделения подготовить отдельную копию слоя с увеличенным контрастом — так будет проще «отловить» дефекты. В дальнейшем эту копию можно будет удалить, используя лишь сделанное в ней выделение. Также, если речь идет о ретуши цветного снимка, то контрастную копию следует обесцветить. Не следует переживать из-за того, что границы выделенных дефектов будут слишком резкие — мы это сейчас поправим.



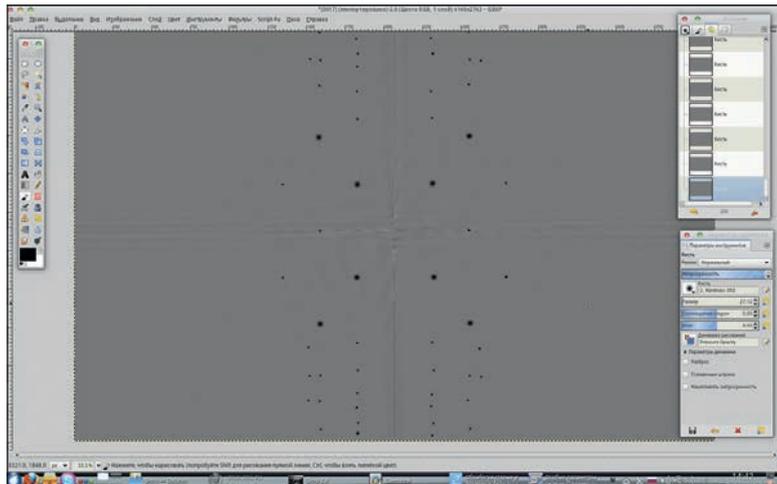
► Рис. 2. Смещаем выделение в сторону, а затем копируем «здоровые» участки изображения.

Итак, выделение на копии сделано; копию можно удалять. Теперь увеличиваем выделение на 1 пиксель (Выделение > Увеличить) и растушевываем его на ~3 пикселя (Выделение > Растушевать). Далее выбираем инструмент перемещения () и переводим его в режим перемещения выделения (кнопка  в параметрах инструмента). Сдвигаем выделение в любую сторону на такое расстояние, чтобы в зону выделения не попадали дефекты (рис. 2). Теперь копируем содержимое слоя под выделением (Правка > Копировать) и сразу же вставляем его (Правка > Вставить). Меняем режим перемещения обратно, выбрав вариант Слой (кнопка  в параметрах инструмента), и перемещаем

► Рис. 3. Та самая текстура бумаги.



» Не хотите пропустить номер? Подпишитесь на [www.linuxformat.ru/subscribe/!](http://www.linuxformat.ru/subscribe/)



➤ Рис. 4. Так выглядит типичная карта, созданная фильтром FFT Forward для изображений с текстурой. Черным показаны уже замазанные «всплески».

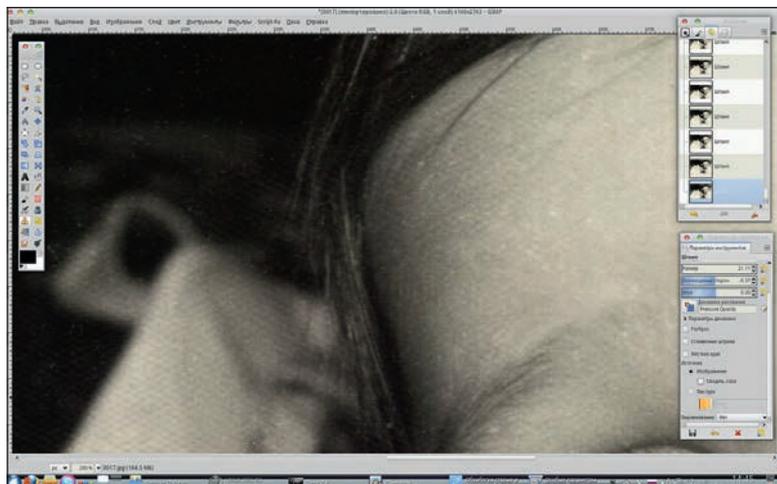
вставленное плавающее выделение обратно на место пыли и царапин. Чтобы правильно «попасть», измените режим наложения у вставленного фрагмента на Умножение: так вы сможете точно совместить фрагмент с дефектными областями. Наконец, можно вернуть нормальный режим наложения и прикрепить фрагменты к слою (кнопка на панели слоев).

Для получения более убедительного результата изначально разделите ваше изображение на светлую и темную части и примените описанный метод отдельно для каждой из частей. Разделить снимок можно так: делаем копию слоя, обесцвечиваем, выделяем по цвету все светлые оттенки, слегка увеличиваем выделение (1–2 пикселя обычно хватает), вырезаем выделение и делаем из него отдельный слой (Правка > Вставить, затем кнопка на панели слоев).

У вас должно получиться три слоя: исходный фон и две части со светлыми и темными оттенками соответственно. Эти два слоя следует использовать для получения правильного выделения, в то время как итоговое копирование «здоровых» фрагментов и их перемещение на место дефектов удобнее делать на исходном фоне.

Разумеется, во время такой процедуры иногда придется корректировать выделенную область — например, с помощью лассо (в режиме добавления или вычитания выделения (кнопки и соответственно).

➤ Рис. 5. Результат работы обратного фильтра FFT Inverse. Текстура практически подавлена.



Долговечность

Считается, что фотографии в цифровом виде долговечнее своих бумажных собратьев, но это скорее теория, а на практике следует как раз не выбрасывать исходные бумажные фото: через несколько лет наверняка появится более совершенная сканирующая техника, и иметь под рукой оригинал не повредит.

БПФ: боремся с текстурой

Пыль пылью, но есть проблема посерьезней: в старые времена было модно печатать фотографии на красивой рельефной бумаге. Обычно такие снимки имеют толстую подкладку из картона и считаются более устойчивыми к износу. Однако при оцифровке рельеф бумаги хорошо виден, и как-либо «замазать» его нельзя. Если рельеф «нерегулярный», напоминающий кракелюры, то тогда выбор невелик: рекомендую отсканировать такое фото 4 раза, каждый раз поворачивая его на 90 градусов. Потом снимки нужно будет аккуратно совместить в GIMP и выбрать нужный режим наложения. Текстура станет менее заметной (этот эффект объясняется тем, что рельеф бумаги по-разному отражает свет от лампы сканера под разными углами). Но вполне возможно, что рельеф именно вашего снимка — регулярный (ромбики, квадратики, одинакового размера бугорки, решетка...), и это значит, что мы сможем воспользоваться быстрым преобразованием Фурье. Пример фото с такой текстурой показан на рис. 3.

Быстрое преобразование Фурье (БПФ — Fast Fourier Transform, FFT) — это один из способов обратимого разложения сигнала на частоты и амплитуды. Применительно к растровому изображению, подобная процедура способна наглядно показать нам повторяющиеся элементы изображения в виде «всплесков» сигнала. Преобразованная по алгоритму БПФ картинка с повторяющимся рисунком текстуры содержит набор таких «всплесков», которые всегда симметричны относительно центра (рис. 4). Закрасив области «всплесков» сигнала и проведя обратное преобразование Фурье (FFT Inverse), можно получить исходное изображение, в котором текстура волшебным образом отсутствует (рис. 5).

Теперь немного значимых подробностей. Вам потребуется дополнительно установить в GIMP модуль Fast Fourier Transform (см. врезку «Где искать модули»), который добавит в пункт Фильтры > Общее два новых фильтра: FFT Forward и FFT Inverse, для прямого и обратного преобразования соответственно.

После прямого преобразования следует обычной кистью небольшого размера замазать области «всплесков» черным или серым цветом (годится любой оттенок). «Всплески» всегда расположены симметрично относительно центра изображения, и вам потребуется закрасить все такие места, за исключением самого центра (это важно!).

Наконец, делаем обратное преобразование (Фильтры > Общее > FFT Inverse) и наслаждаемся результатом!

Повышаем четкость

Не все фотографии идеально сфокусированы на объекте съемки: кое-где можно наблюдать размытость, либо даже размытие движением — результат долгой выдержки или трясущихся рук фотографа. Если такой дефект проявлен сильно, то вряд ли что-то можно будет сделать, но в остальных случаях будет полезно повысить резкость фотографии. Рассмотрим недостаточно резкую фотографию на рис. 6.

Для исправления резкости в ней есть несколько отдельных инструментов. Я сознательно не буду здесь писать про уже

➤ Пропустили номер? Узнайте на с. 108, как получить его прямо сейчас.



➤ Рис. 6. Отсканированный снимок обычно слегка «замылен», ему не хватает резкости...



➤ Рис. 7. Фильтр High Pass с правильно подобранным радиусом может заметно улучшить резкость.

набившую оскомину нерезкую маску или обычный фильтр типа Sharpen, который повышает контрастность соседних пикселей. Есть кое-что поинтереснее.

» High Pass

Для редактора *GIMP* есть замечательный дополнительный модуль High Pass — «фильтр верхних частот», который умеет повышать резкость фотографии, не увеличивая при этом цветовой шум. Модуль в работе создает дополнительный слой и отображает на нем карту нерезкости исходного снимка. Однако, в отличие от классической «нерезкой маски», слой с этой картой мы наложим на фон в режиме Рассеянный свет (выберем на панели слоев в меню Наложение).

Вы сразу заметите, что края объектов на снимке стали резче (рис. 7). Применяя регулировку уровней к слою с картой нерезкости, можно менять величину воздействия фильтра. Чем контрастнее карта, тем больше высокочастотное усиление и «острее» снимок.

На практике High Pass имеет смысл применять, если исходный снимок изначально блеклый или выцветший — полученный результат вас приятно порадует.

» Refocus

Этот интересный модуль существует с 2003 года и уже был описан в нашем журнале (*LXF112*). Поэтому стоит здесь рассказать о некоторых нюансах в его использовании при работе с архивными фотографиями.

Refocus пытается восстановить «фокусировку» снимка, используя т. н. «матрицу свертки». При правильном подборе параметров (радиус, размер матрицы и т. п.) данный фильтр прекрасно устраняет «замыленность», вызванную неверной фокусировкой объектива, однако он почти бессилен перед размытием движением — то есть если дефект вызван слишком долгой выдержкой,

то фильтр не выдаст хороший результат. Однако в нашем случае Refocus будет полезен как раз для своего основного назначения: мы уберем легкую «замыленность», которая всегда присутствует у отсканированных изображений. Для этого достаточно применить фильтр один раз со стандартными параметрами. При желании эффект можно усилить повторным применением.

Вместе с восстановлением фокуса на фотографии проявятся шумы и артефакты, поэтому здесь уместно дать советы:

➤ Фильтр лучше применять уже после цветокоррекции — в противном случае могут вылезти нежелательные детали, снимок станет «грязным».

➤ Фильтр обязательно нужно применять к несжатому файлу. Иными словами, картинка JPEG плохо подходит, т. к. фильтр усилит ее блочность.

➤ Творческое применение Refocus — сознательная установка слишком «сильных» параметров, которые сделают снимок очень резким и помогут тем самым обнаружить все мелкие пылинки и дефекты. В дальнейшем их можно будет «точечно» заретушировать на исходном слое и получить в итоге аккуратный результат.

«Сначала манипуляции со светом, затем с текстурой или резкостью.»

крупные пылинки и дефекты. В дальнейшем их можно будет «точечно» заретушировать на исходном слое и получить в итоге аккуратный результат.

Напоследок...

Есть несколько общих рекомендаций при фоторетуши в *GIMP*. Это касается как фотографий вообще, так и старых/поврежденных снимков:

При бережном отношении к качеству результата имеет значение порядок действий. Сначала нужно произвести все манипуляции со светом, затем — с текстурой или резкостью.

Черно-белые снимки нужно обесцвечивать после сканирования (Цвет > Обесцвечивание) и лишь затем ретушировать: любая цветность здесь имеет паразитную природу. Для создания приятной сепии лучше всего под конец слегка тонировать фото (Цвет > Тонирование).

При необходимости «замазать» дефекты вручную не спешите использовать стандартный инструмент Штамп: иногда гораздо лучше подготовить копию слоя с примененным фильтром Удаление пятен (Фильтры > Улучшение) и заимствовать «здоровые» участки из него. Метод такой: слой с удаленными пятнами кладем вниз, а исходный слой — вверх, не забыв добавить к нему альфа-канал. После этого мягкой кистью удаляем дефекты при помощи инструмента Ластик. Таким способом очень удобно удалять трещины на фоне или малозначимых частях снимка. При ретуши лица без кропотливой работы обычным Штампом скорее всего не обойтись. **LXF**

Где искать модули

Если какого-нибудь из описываемых фильтров в вашей системе нет, то первым делом стоит открыть пакетный менеджер и поискать по шаблону `gimp-plugin`. Для многих дистрибутивов есть пакет `gimp-plugin-registry`, который включает богатый набор всевозможных расширений для *GIMP*. Если же вы решили пойти путем джедая и все сделать вручную, то не забывайте, что по умолчанию *GIMP* хранит подключаемые модули (plugins) тут: `/usr/lib64/gimp/2.0/plugin-ins/`.

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 Сбой с установкой обоев
- 2 Клонирование готовой системы
- 3 Печать файлов man/info
- 4 Проблемы с разбиением диска
- 5 Понимание LVM
- 6 Безопасность браузера

1 Переоформление KDE

В Я написал простую программу на Python, которая каждый день загружает случайное изображение с web-сервиса и устанавливает его в качестве обоев. Все замечательно работает на ноутбуке с Fedora 18 (Gnome) и на компьютере с Windows 7, которым я пользуюсь на работе. Однако на домашнем компьютере стоит KDE 4.9.5, и я не могу разобраться, как установить обои командой оболочки или с помощью Python. Или в KDE 4 это невозможно?

Нил Бэйли [Neal Bailey]

О Как и со многими другими настройками в KDE, Вам должен был помочь вызов D-BUS (в KDE 3 это был вызов DCOP), но данная функция еще не реализована. Сейчас выпущен 11-й релиз KDE 4, но похоже, что эта функция пока не ходит в главных (хорошо уже и то, что нам все-таки вернули кнопку выключения компьютера).

Однако способ сделать это есть, и он будет прекрасно работать из любого скрипта или исполняемой программы. Загрузите плазмод *Scripted Wallpaper* с kde-apps.org (<http://kde-apps.org/content/show.php/Scripted+Wallpaper?content=156317>), распакуйте архив и запустите команды

```
cd Scripted_Wallpaper
./install.sh
```

Если Вы запустите скрипт от имени своего пользователя, у Вас спросят пароль, так как для установки нужен **sudo**. Теперь откройте окно Настройки рабочего стола [Desktop Settings], и в разделе выбора обоев появится вариант *Scripted Wallpaper*. Укажите путь до скрипта или программы, ее аргументы и периодичность запуска. Также можно указать картинку для показа на случай ошибки скрипта, каковая всегда может произойти при загрузке из Интернета.

Теперь нужно сделать так, чтобы скрипт передавал путь к новому изображению в стандартный вывод. Плазмод будет запускать скрипт через заданные интервалы времени, считывать путь и устанавливать новое изображение в качестве обоев.

В последующих строках скрипт может выводить другие параметры, например, режим масштабирования и цвет фона. Вот пример скрипта, который загружает астрономическую картинку дня NASA и выводит информацию, необходимую для ее установки в качестве обоев с масштабированием и черным фоном:

```
#!/bin/sh
image_dir=/home/me/wallpapers
image=$(wget -q http://apod.nasa.gov/apod/astropix.html -O - | grep -m 1 -o 'image.*\.jpg')
wget -q -N -P $image_dir http://apod.nasa.gov/apod/$image
echo $image_dir/${basename $image}
echo "ScaledResize"
echo "0 0 0"
```

2 Клоны и установка

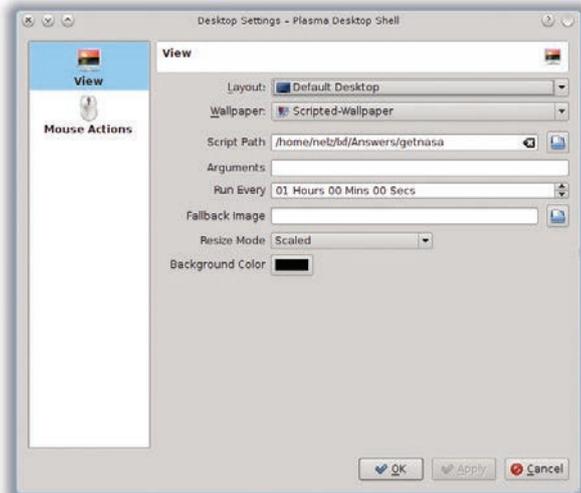
В Я хочу создать главную версию системы на базе Debian и затем клонировать ее на новые компьютеры по мере необходимости. Так как я хочу делать это раз или два в месяц, мой план таков:

- 1 Создать основную систему на виртуальной машине, которую я могу поддерживать обновленной.
- 2 Загрузить новый компьютер с Live CD, разбить диски, настроить RAID и затем перенести главную систему на новый компьютер с помощью *rsync*.
- 3 Перезагрузить новый компьютер и внести все мелкие доделки в настройки.

Так как новые компьютеры предназначены для конкретных задач, обновления им не нужны, за исключением обычных через *apt-get*. Меня беспокоит то, что на сервере виртуальной машины — процессор AMD, а на новых компьютерах скорее всего будут процессоры Intel; не вызовет ли это проблем?

Кевин Голдинг [Kevin Golding]

О Пакеты компилируются под общую спецификацию процессора, для 32-битных дистрибутивов это обычно i686. Для 64-битных дистрибутивов различия между процессорами Intel и AMD в отношении их наборов команд минимальны. Если Вы не запускаете некое особым образом скомпилированное ПО — скажем, супер-элитный дистрибутив Gentoo «для лихачей», учитывающий нюансы процессора — Вы не должны столкнуться ни с какими проблемами совместимости.



Scripted Wallpaper позволит изменить обои в KDE в любой момент.

Более вероятны проблемы с виртуализированными устройствами, особенно графическими драйверами; поэтому, чтобы все работало правильно, может потребоваться небольшая дополнительная настройка — возможно, установка нескольких дополнительных пакетов.

У Вас также появятся проблемы с именованнием устройств сети из-за постоянных имен устройств *udev*. Оно связывает имена сетевых адаптеров с MAC-адресами, и если у Вас больше одной сетевой карты, они обязательно получат одни и те же имена устройств.

Так как MAC-адрес Вашего нового компьютера будет отличаться от адреса виртуальной системы, сетевой интерфейс компьютера получит имя *eth1*, так как *eth0* уже занято. Решение — удалить файл */etc/udev/rules.d/70-persistent-net.rules* из клонированной системы; в основной системе удалять его нет смысла, поскольку он все равно опять создастся при очередной перезагрузке. Без этого файла новая система будет считать, что все адреса доступны, и назовет Вашу карту *eth0*.

Кстати, в свежей *udev* используется совершенно другой метод именованвания, который исключает необходимость постоянных имен и вызовет на форумах бесконечные вопросы типа «почему моя сетевая карта теперь называется *enp5s0*».

Вместо удаления файла можно добавить аргумент *exclude* к команде *rsync*. Могут быть и другие файлы, которые Вы хотите опустить при копировании, например, */var/log/**; перечислите их в файле по одному на строке, укажите этот файл в *rsync*, и команда исключит их все.

Если Вы запускаете *rsync* из основной системы, Вам нужно знать о виртуальных каталогах,

таких как `/dev` и `/sys`. Их можно исключить с помощью параметра `--one-file-system`, но в каталоге `/dev` корневого раздела могут быть файлы, которые Вам необходимы.

Одно из решений — монтировать корневую файловую систему в другом месте и синхронизировать ее через `rsync`, так мы уберем обе эти проблемы:

```
mkdir -p /mnt/root
mount --bind /mnt/root
rsync --exclude-from /rsync-excludes /mnt/root
root@newcomputer:/
umount /mnt/root
```

Другой вариант — в зависимости от используемой системы и формата виртуального диска можно смонтировать виртуальный диск с незапущенной виртуальной машиной и выполнить синхронизацию с `rsync` из основной системы.

3 Печать man-страниц

Взовите меня старомодным, но я люблю печатать все на бумаге. На страницах `man` и `info` очень много информации, но я нашел печатную версию более удобоваримой. Я нашел файлы `man`-страниц в своей системе, у них расширение `.gz`, поэтому я попробовал распаковать их с помощью `Gunzip`, но теперь не понимаю, что с ними делать. Видимо, они написаны на каком-то языке разметки, но как преобразовать его в простой текст для печати?

Джон Кэкстон [John Caxton]

Омы в *Linux Format*, разумеется, не имеем ничего против тех, кто предпочитает печатную информацию. `Man`-страницы созданы с использованием системы форматирования `troff`, но их можно конвертировать в другие форматы самой командой `man`. По умолчанию `man`-страницы обрабатываются Вашей обычной программой просмотра (обычно `less`), но если указать параметр `-t`, страница форматируется и выводится в PostScript (это язык, на котором говорит

система печати). Скажем, `man`-страницу `Bash` можно напечатать так:

```
man -t bash | lpr
Выход в PostScript также можно перенаправить в другие программы — например, создать PDF-версию командой
man -t bash | ps2pdf - bash.pdf
```

Или можно украсить вывод с помощью `a2ps`. Чтобы напечатать две страницы рядом друг с другом на одной странице бумаги с названием документа вверху в центре, скомандуйте `man -t bash | a2ps --columns=2 --centertitle="Bash" | lpr`

По-моему, это удобный формат: текст остается читаемым, но количество физических страниц уменьшается.

Со страницами `info` все немного иначе, потому что они представляют собой набор страниц, связанных гиперссылками, а не линейный текст `man`. Один из вариантов — воспользоваться параметром `--subnodes`, с которым все страницы документа сводятся в один поток. Это обычный текст, предназначенный для чтения на экране, но так как "а" в `a2ps`, которой мы только что воспользовались для преобразования в PostScript, означает "any [любой]", его можно воспользоваться и здесь: `info --subnodes bash 2>/dev/null | a2ps --columns=2 --center-title="Bash" | lpr`

4 Ошибки при разбиении

Впользуясь рекомендациями из `LXF166` и руководством по `dd`, я загрузился с `Live DVD Mint 14` и клонировал 1-ТБ диск с помощью `dd`. Таблицы разделов на дисках были разными, и с помощью `lsblk` я легко определил правильные пути для команды `dd`. Но когда `dd` закончила работу, `lsblk` показала новый диск со старой таблицей разделов. `Gparted` показала правильную таблицу разделов, но сказала, что на диске нет данных. После выключения системы и загрузки с нового диска оказалось, что все хорошо, и новый диск, по всей видимости, является дубликатом

Терминалы и суперпользователи

Мы часто предлагаем в качестве решения проблемы ввести те или иные команды в терминале. Хотя обычно то же самое можно сделать с помощью графических утилит дистрибутива, такие решения будут слишком конкретными (будут зависеть от дистрибутива). Команды в терминале более гибкие и — самое главное — ими можно пользоваться во всех дистрибутивах. Команды настройки системы часто нужно выполнять от имени суперпользователя, называемого также `root`. Существует два основных способа это делать, в зависимости от используемого дистрибутива. Во многих дистрибутивах, особенно в `Ubuntu` и его производных, перед командой можно написать `sudo` — при этом будет запрошен пароль пользователя, и ему будут предоставлены привилегии `root` только на время выполнения команды. В других дистрибутивах применяется команда `su`, для использования которой требуется ввести пароль `root` и которая предоставляет полный доступ `root` до того момента, пока вы не наберете `logout`. Если в вашем дистрибутиве используется `su`, запустите ее один раз и выполняйте любые заданные команды без предшествующей `sudo`.

оригинала. Почему `lsblk` и `Gparted` дали мне неверную информацию?

Номер `LXF166` также вдохновил меня на перенос системы `Windows` в `VirtualBox`. С помощью `dd` (как и предлагалось в статье) я создал файл `IMG` на пустом диске, но во время преобразования в формат `VDI` диск переполнился. Когда я удалил файлы `IMG` и `VDI` и опустошил корзину, диск все еще был полон. Чтобы вернуть обратно пространство на нем, пришлось реформатировать »



Коротко про...

Выделение памяти

Вы вдруг обнаружили, что после пары часов работы системы осталось всего несколько килобайт свободной оперативной памяти? Возможно, вы думаете, что нужно прикупить еще памяти, но на самом деле, независимо от объема вашей физической памяти, через несколько часов после загрузки, а то и раньше, она будет использоваться полностью.

Видимо, это чудовищная ошибка, и где-то в вашем любимом дистрибутиве налицо утечка памяти? Ответ — нет, ядро просто использует память по максимуму. Хотя использование лишней памяти — это трата ресурсов, простаивание памяти — еще худшая трата ресурсов, поэтому

ядро `Linux` выделяет память, не используемую активно, на такие вещи, как кэши файловой системы, чтобы повысить производительность. Убедиться в этом можно, взглянув на параметр "buffers [буферы]" в выводе команды `top`:

```
Mem: 1028092k total, 1018704k used, 9388k free, 49760k buffers
```

Эта строка показывает, что из 1 Гб оперативной памяти свободно менее 10 Мб. Почти половина памяти используется как буферы, которые значительно ускоряют доступ к диску, кэшируя записанные данные в памяти и считывая больше, чем нужно, в ожидании следующего чтения. Если эта память становится

нужна для чего-нибудь поважнее, она незамедлительно освобождается.

В памяти остаются не только данные файловой системы. Попробуйте проверить свободную память, затем запустить и остановить большую программу вроде `Firefox` — объем свободной памяти уменьшится. Теперь снова запустите `Firefox` и посмотрите, насколько быстрее он стал. Это произошло потому, что ядро продолжает хранить данные в памяти для последующего использования. То же относится к разделяемым библиотекам, поэтому программы из одной и той же рабочей среды могут работать заметно быстрее.

его. Затем переполнился мой корневой раздел в 100 ГБ (на другом диске). Чтобы исправить это, понадобилось переформатировать корневой раздел и переустановить систему. Можете предположить, что же произошло, и как избежать этого снова, когда я еще раз попытаюсь виртуализировать свою систему Windows?

Коллин Грэм [Colin Graham]

Ядро считывает таблицу разделов при первом обнаружении устройства — либо при загрузке, либо при его последующем подключении. Если Вы вносите изменения в разбивку диска, нельзя быть уверенным, что ядро использует самую последнюю конфигурацию. Если диск используется — а для этого достаточно смонтировать раздел, и не обязательно обращаться к диску — у ядра может не получиться обновить информацию о разделах у себя. Многие программы, в том числе *lsblk*, считывают информацию ядра из */sys*, и поэтому возвращают неактуальные данные. Сделать так, чтобы у ядра точно была актуальная информация, можно тремя способами. Если диск подключен через USB, размонтируйте и отключите его; когда Вы подключите его снова, считается свежая таблица разделов. Для внутреннего диска попробуйте запустить команду `sudo hdparm -z`

Эта команда принудительно перечитывает таблицу разделов, а при неудачном завершении выводит соответствующее сообщение. Третий вариант, который работает всегда — перезагрузка.

Что касается фантомных файлов — к сожалению, отформатировав раздел, Вы уничтожили почву для умозаключений, но несколько вариантов есть. Один из возможных — файловая система была повреждена, когда место на диске закончилось, поэтому корзину очистить не удалось. Чтобы такое исправить, обычно достаточно размонтировать файловую систему — если это

системный раздел, Вам понадобится Live CD или загрузочная флэшка — и запустить для нее команду `fsck`:

```
sudo fsck -f /dev/sdaX
```

Она должна исправить все ошибки и даже удалить частично записанные файлы — или, по крайней мере, позволить Вам это сделать. В Вашем случае корневая файловая система не должна была пострадать. Если она все же была затронута, в системных журналах будет множество сообщений об ошибках доступа и каталог */var* будет заполнен. Тогда лучше всего перезагрузиться с Live-дистрибутива и очистить лог-файлы.

Есть несколько утилит, которые помогут понять, куда исчезло некогда обширное дисковое пространство. Из графических программ моя любимая — *Filelight* для KDE, а для Gnome есть *Disk Usage Analyser* или команда `du` в терминале. Посмотреть, куда исчезло свободное место, можно командой

```
sudo du -schx /
```

Найди алчного пожирателя свободного места, например, */var*, ограничьте поиск командами:

```
sudo du -schx /var
```

```
sudo du -schx /var/log
```

и т. д. В Linux очень мало проблем, для решения которых требуется форматирование или переустановка системы. Если не сметать сразу все, а поискать причину проблемы, это упростит ее решение, если она снова возникнет.

5 Стандарты LVM

Есть ли стандартный способ работы с дисками в LVM? Некоторые компании сначала разбивают диск на разделы, а потом превращают их в физические тома:

```
pvccreate /dev/sda1
```

Другие превращают в физические тома неразбитые диски:

```
pvccreate /dev/sda
```

Каков стандартный способ? Есть ли причина, по которой не стоит превращать неразбитые диски в физические тома?

Бернард Флак [Bernard Flach]

Стандартного подхода нет, ведь главное достоинство LVM — гибкость; поэтому используйте любой способ, наиболее подходящий к Вашей ситуации. Физические тома можно создавать почти на любом блочном устройстве, включая разделы и целые диски, а также и другие блочные устройства, например, блочные устройства, зашифрованные *dm-crypt*, программные устройства RAID и даже файлы локальной петли на обычных разделах (хотя обычно это делается только для тестирования).

Причин не разбить целый диск нет — на самом деле, я задал бы противоположный вопрос: если Вы хотите отдать весь диск под группу томов, зачем сначала разбивать его? Если Вы будете пользоваться диском полностью, на нем не должно быть таблицы разделов. Стереть любую существующую таблицу разделов можно командой:

```
sudo dd if=/dev/zero of=/dev/sdX bs=512 count=1
```

Итак, настраивайте свои логические тома, исходя из собственного удобства. Единственный важный момент, который полностью игнорируют некоторые установщики дистрибутивов, в том, что логические тома должны быть такого размера, какого Вам необходимо. Логический том легко сделать и меньше, и больше, но файловую систему проще именно увеличить. Поэтому сделайте их, например, величиной с желаемый размер плюс 40–50 % на то, чтобы новые большие файлы записывались без фрагментации, а остальное место останется доступным до того момента, когда Вам понадобится.

5 Попытки доступа

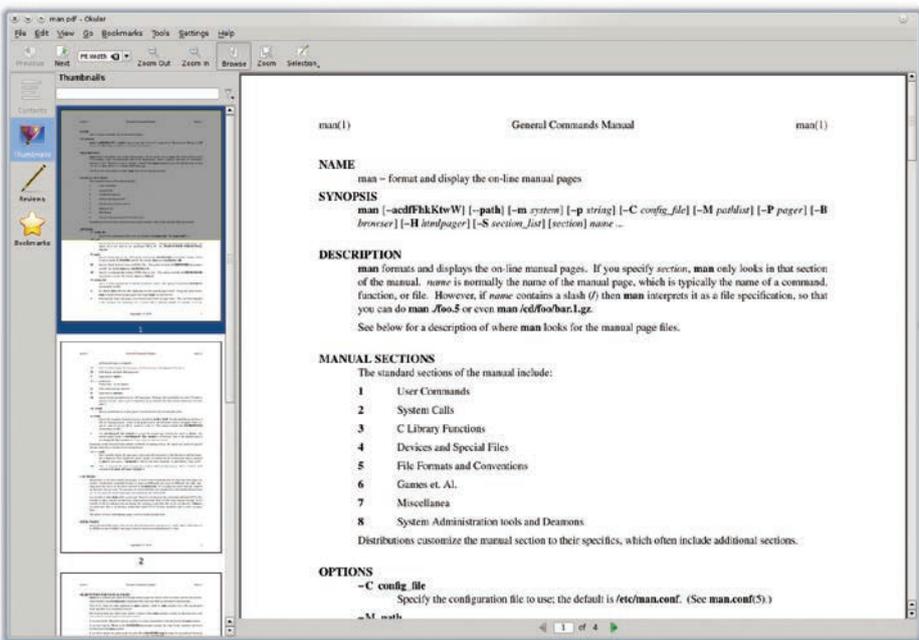
Время от времени *Firefox* сообщает, что доступ к моему компьютеру пытается получить другой компьютер, и предлагает разрешить или запретить его. Я всегда запрещаю доступ. Это достаточная защита, или я иду на излишний риск? Я начал замечать, что попытки доступа возобновились, и все они поступают с сайта spitfireuk.net. У меня все еще Ubuntu 10.04.

Питер Рэтклифф [Peter Ratcliffe]

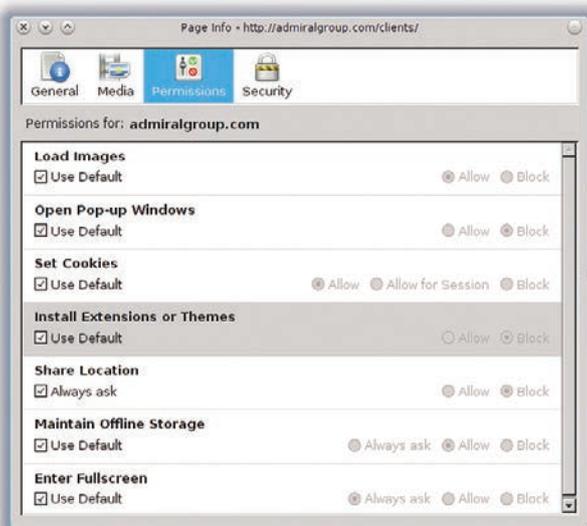
На такие сообщения нужно обращать внимание всегда, если Вы не можете их объяснить. Однако Вы не указали точный текст сообщения, а это важно. *Firefox* не должен никак участвовать в удаленном доступе к Вашему компьютеру, если только Вы не установили расширение, которое разрешает такое, и *Firefox* предупреждает Вас об этом, поэтому проверьте свои расширения и отключите все, которые несут потенциальный риск. Также отключите все расширения, без которых можете обойтись.

Можно даже попробовать запустить *Firefox* с командной строки с отключенными темами и расширениями:

```
firefox -safe-mode
```



Страницы `man` и `info` можно печатать и сохранять в других форматах — например, в PDF.



▶ Права доступа на уровне сайта в Firefox могут улучшить безопасность и приватность.

Другой вариант — Вас предупреждают о том, что сайт пытается сохранить или прочитать данные с Вашего компьютера через cookie. Куки обычно ограничены одним сайтом, то есть куки, сохраненные dodgystuff.com, должны считываться только при загрузке контента с dodgystuff.com. На spitfireuk.net размещено много доменов и сайтов, поэтому то, что Вы видите, может быть всего лишь программой для отслеживания страниц в действии. То, что мы говорим «всего лишь», отнюдь не обязательно означает, что мы одобряем

это — только Вы можете принимать решения о своей безопасности; но это не похоже на попытку взлома Вашего компьютера. Управлять этим можно в разделе Advanced [Дополнительные] окна настроек Firefox. Зайдите на вкладку Network [Сеть] и убедитесь, что галочка “Tell me when a website asks to store data for offline use [Сообщать мне, когда веб-сайт просит сохранить данные для автономной работы]” установлена — раз уж Вы видите эти сообщения, она и должна быть установлена. Зайдите на страницу, которая выдает предупреждение, затем откройте Tools > Page Info (Ctrl-I) [Инструменты > Информация о странице (Ctrl-I)] и перейдите на вкладку Permissions [Разрешения]. Здесь можно изменить параметры установки куки или автономного хранения данных. Также можно зайти на вкладку Security [Безопасность], чтобы просмотреть или удалить существующие куки.

Кстати, когда Вы будете это читать, поддержка Ubuntu 10.04 закончится, то есть обновлений безопасности больше выпускаться не будет. Если Вы хотите продолжать работать в Сети без проблем, серьезно подумайте об обновлении системы до более свежего релиза LTS. **LPF**

Помогите нам помочь вам

Ежемесячно мы получаем несколько писем, на которые не в состоянии ответить, поскольку 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-файле, который вы сможете приложить к своему письму.

Альтернативный и не уступающий в удобстве вариант — *Ishw* (<http://ezix.org/project/wiki/HardwareLiStEr>). Одна из указанных программ непременно должна быть включена в ваш дистрибутив (а иногда 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

LDAP

▶ **Опять эти ваши аббревиатуры... как она расшифровывается?**

LDAP — сокращение от “Lightweight Directory Access Protocol [облегченный протокол доступа к каталогам]”.

▶ **Как обычно, яснее не стало. Что это такое?**

LDAP — протокол для обслуживания и изменения информации в каталогах по сети.

▶ **Нечто вроде «Желтых страниц» в сети?**

Вроде. Каталог LDAP может представлять собой любой набор логически и иерархически организованных данных. Самый известный пример — телефонная книга, которая состоит из имен, адресов и телефонов, отсортированных в алфавитном порядке. Или такие справочники, как «Желтые страницы».

▶ **То есть это похоже на телефонную книгу в сети?**

Это может быть телефонная книга, а может быть любой другой набор структурированных данных: например, компьютеры в сети, продукты и цены, каталог документов... на самом деле, почти все что угодно, подающееся структурной организации. Интернет тоже не обязателен. Большинство серверов LDAP работают в частной сети или на одном компьютере. В конце концов, вы вряд ли захотите, чтобы себестоимость продукции вашей компании была доступна в Интернете.

▶ **Похуже на то, что каталог LDAP — своего рода база данных?**

Да, один из видов. LDAP не предназначен для сложных наборов данных и запросов, поэтому он называется «облегченным». По той же причине PostgreSQL или Oracle из-

быточны для хранения списка сотрудников офиса и их добавочных номеров. Для решения таких задач предназначен LDAP.

▶ **Поэтому его гораздо проще настроить?**

Ну, да, если проделать это несколько раз. LDAP — не база данных SQL, и вам придется изучить новый подход; но все хорошо задокументировано, и есть множество примеров схем.

▶ **Стоп! Что такое схема?**

Схема определяет структуру каталога и атрибуты, которые может содержать каждая запись. Это аналог определения таблицы в SQL.

▶ **Когда у меня появится каталог LDAP, как им можно будет пользоваться? Нужны ли специальные программы?**

Вовсе нет. У многих популярных программ есть возможность использовать каталог LDAP в качестве источника информации. Например, многие почтовые клиенты могут применять LDAP для адресных книг.

▶ **Чем плохи внутренние адресные книги, которые обычно используются?**

Они внутренние, а следовательно, работают только в пределах данного почтового клиента. В общем каталоге почтовые клиенты смогут искать адреса, Samba — аутентифицировать пользователей; каталогом также смогут пользоваться многие другие программы. Это единственный централизованный источник данных, которые можно использовать так, как вам требуется (или — как решит системный администратор).



LXF HotPicks



Майк Сондерс

Издавив самые недостижимые и укромные уголки Интернета, Майк точно знает, где таятся главные сокровища открытого кода.

LIVES » Sxiv » G4L » ETM » Plgen » Catfish » Siege » Colorful » Choria » Gitg » Atratus

Видеоредактор

LIVES

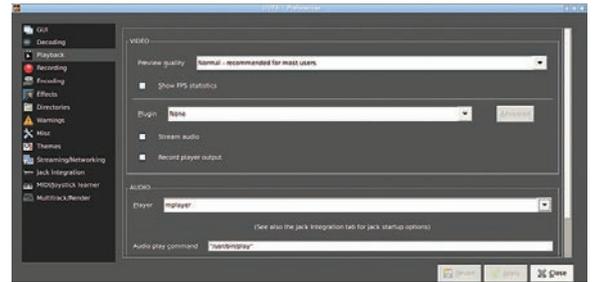
Версия: 2.0.2 Сайт: <http://lives.sourceforge.net>

LiVES за последние годы несколько раз появлялся в HotPicks — его последнее появление датируется LXF145. Причина, по которой мы столь недолго опять обратились к нему — быстрый темп разработки: последняя рассмотренная версия была 1.4.2, но недавно уже увидела свет серия 2.0. Хотя команда LiVES описывает 2.0 как «версию разработчиков», они тут же заявляют, что она «рекомендуется большинству пользователей». Мы считаем это весьма практичным

решением — 2.0 должна быть достаточно стабильна для большинства задач, однако в ней могут быть ошибки, ведь код полностью новый.

С точки зрения зависимостей LiVES требует как минимум *GTK3* для интерфейса, *ImageMagick* для работы с изображениями и *MPlayer* для работы с разными видеоформатами. Если вы установите дополнительные рекомендованные инструменты и библиотеки, такие как *libtheora*, у вас будет больше гибкости при импорте и экспорте клипов.

LiVES предлагает несколько интерфейсов на выбор: так, режим Clip Editor, показанный



► В Preferences вы можете указать LiVES, какие внешние инструменты брать для воспроизведения и конвертирования аудио и видео.

на экранном снимке внизу, предназначен для основных задач редактирования, когда вы работаете с одним файлом и хотите порезать его, добавить эффекты и заново закодировать в новый формат. Далее, имеется режим Multitrack, который значительно сложнее — он предназначен для сессий редактирования нескольких видео- и аудиоклипов. Нам показалось, что оба интерфейса несколько шероховаты, и им не помешала бы некоторая шлифовка, но они не отвлекают от выполнения текущей задачи.

Клипы импортируются с DVD или с видеокамеры через FireWire, и можно даже делать записи с других окон на вашем рабочем столе (очень полезно при создании руководств). Здесь имеется большой выбор фильтров эффектов, от стандартных фильтров для изменения яркости и контрастности до добавления наложения изображений и до сглаживания зернистости рамок.

LiVES растет

По сравнению с версией 1.4.2, LiVES значительно усовершенствован: добавлены новые эффекты, сделаны оптимизации и устранено немало ошибок. Многие форматы, на чтение которых при запуске раньше уходила прорва времени (например, WebM), теперь просматриваются мгновенно, и была реализована поддержка плагина LADSPA. Огромная часть кода переработана, так что если вы пытались работать с LiVES раньше и у вас были проблемы, стоит дать ему еще один шанс. А если вы только подумываете о том, чтобы попробовать свои силы в редактировании видео, он определенно заслуживает внимания.

«Клипы импортируются с DVD или с видеокамеры через FireWire.»

Исследуем интерфейс LiVES

Меню

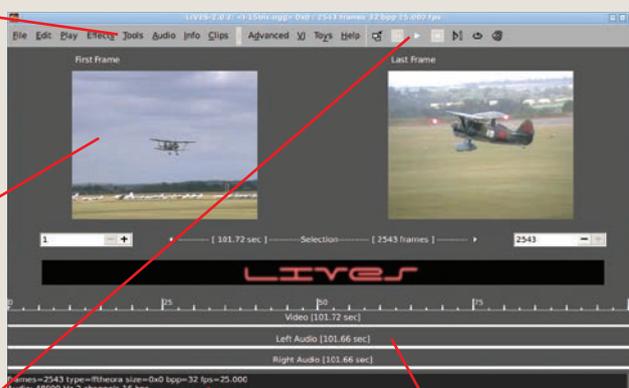
Здесь можно импортировать и экспортировать клипы и применять любой из массы имеющихся эффектов.

Предпросмотр

Правое и левое изображения показывают первый и последний кадр соответственно.

Панель инструментов

Кнопки на ней позволяют воспроизвести весь клип или последнюю выделенную часть.



Вывод

Поглядывайте на эту панель — она отображает ценную информацию из плагинов и внешних инструментов.

Выделение

Щелкайте и перетаскивайте панели, чтобы изменить выделенную длительность видео.

Программа просмотра изображений

Sxiv

Версия: 1.1 Сайт: <https://github.com/muennich/sxiv>

По этому пути проходят многие линуксоиды, и он ведет к просвещению в плане применения клавиатуры. Обычно начинается все с командной строки, с осознания, что приказы компьютеру командами, а не щелчками по бесконечным диалоговым окнам, экономят массу времени. А еще есть управляемые с клавиатуры менеджеры окон, и, конечно, *Vim* и *Emacs*, (бес)славные текстовые редакторы, требующие от пользователя знания множества сочетаний клавиш и команд.

Привыкнув к работе исключительно на клавиатуре, вернуться назад к неудобной, неуклюжей мыши или трекпаду будет сложно. И вы начинаете искать приложения, соответствующие вашему новому способу работы, чисто клавиатурному. Отличный пример — *Sxiv*; это не самый революционный инструмент, а всего лишь программа просмотра изображений, но она заботится о любителях погреть руки клавиатурой. В частности, она предназначена

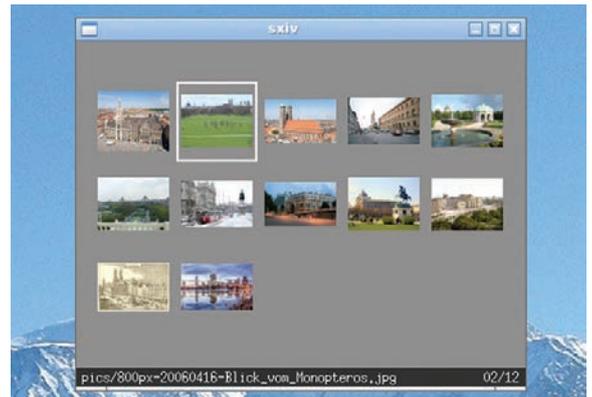
для поклонников *Vi(m)*. Запустите *Sxiv*, указав ей на ряд файлов:

```
sxiv path/*.jpg
```

Она откроет все файлы JPEG в папке **path**. Вы можете следить за строкой состояния внизу экрана, где отображается имя файла, процент масштаба и количество текущих файлов в списке (например, 1/6). Клавиши N и P переключат на следующее и предыдущее изображения соответственно. Для удаления изображения из списка нажмите на D. Поворачивать изображение можно клавишами < и >, увеличивать и уменьшать масштаб — клавишами + и -, а настраивать сглаживание с помощью A.

Просматривая ряд изображений, можно нажать на Enter и переключиться на режим

«Привыкнув к клавиатуре, вернуться назад сложно.»



► Здесь нет никаких панелей инструментов, меню или виджетов — только чистый кайф от работы с клавиатурой.

эскизов, как показано на экранном снимке. Здесь вступают в игру сочетания клавиш в стиле *Vi*: H, J, K и L. Есть уйма других операций с клавишами для переворачивания, горизонтальной прокрутки и выбора масштаба изображений, и все они описаны в руководстве программы (**man sxiv**).

В общем и целом, *Sxiv* — довольно простая программа, но она демонстрирует, насколько быстрее вы можете работать, не прибегая к мыши или трекпаду. Если вы проводите много времени в командной строке и периодически вам надо просмотреть изображения, но неохота приглашать грузунов, ее нельзя пропустить.

Утилита клонирования разделов

G4L

Версия: 0.44 Сайт: <http://sourceforge.net/projects/g4l/>

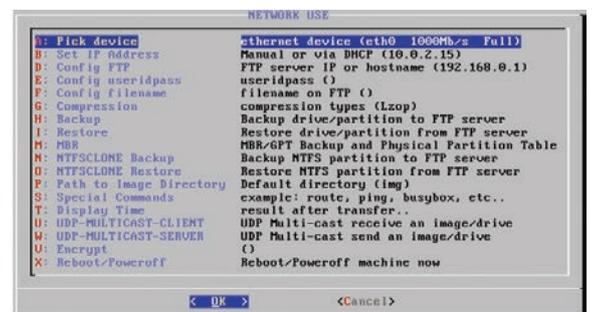
Создать резервную копию целой установки Linux — задача не из простых, если вы приметесь за нее голыми руками. Вам придется сохранить свои личные файлы, а затем файлы настройки, и стартовые скрипты, которые вы меняли, и все, что есть в **/var/lib**... и нет этому конца. Более простой подход — создать образ всего жесткого диска, то есть скопировать все содержимое жесткого диска в файл. А затем вы можете выполнить рискованную операцию, например, обновление дистрибутива на месте, и если все вдруг сломается, вы вернете образ назад на жесткий диск, и все снова будет как было.

Ну, а если у вас только один жесткий диск? Где хранить файл образа? Для подобных ситуаций нужен инструмент создания загружаемой резервной копии диска, который может хранить образы диска в сети — что и делает утилита *G4L*. Она весит менее 100 МБ, и после того, как

вы запишете ее на CD-R и загрузите его, перед вами появится интерфейс, работающий через меню, очень похожий на программы установки конца 1990-х, когда еще не все обратилось в графическое.

Для операций по резервному копированию диска и восстановления у вас есть три режима. *Network Use* позволяет сохранять образ диска в сети через FTP (или восстанавливаться с него). Вы можете выбрать формат сжатия для образа и при необходимости включить шифрование. В *Local Use* вы можете сделать резервную копию одного жесткого диска и сохранить образ в директорию на другом, или сохранить его в удаленную директорию, которая была подмонтирована с помощью SSH, NFS

«Если все вдруг сломается, вы вернете образ на жесткий диск.»



► Не-технарей изобилие опций может обескуражить, но настройки по умолчанию, как правило, отлично работают.

или CIFS. Вы можете делать восстановление и из этого места. И еще есть “Click ‘n Clone”, обеспечивающий прямое копирование одного диска на другой.

G4L предполагает, что вы уже достаточно продвинутый пользователь Linux и не страшитесь большого количества опций и технической информации. Она позволяет давать разные команды администрирования для настройки сети и получения информации о разделе изнутри интерфейса, работающего на меню.

Итак, когда вы в следующий раз решите сделать **apt-get dist-upgrade** или **raspm -Su**, используйте *G4L*, чтобы сделать моментальный снимок своего жесткого диска.

Менеджер задач и событий

ETM

Версия: 2.1.3 Сайт: <http://people.duke.edu/~dgraham/etmq/>

Если только у вас не фантастический КПД, или у вас всего и занятий, что читать этот журнал (тогда вы счастливы), то у вас определенно должен быть список дел. И есть вероятность, что составлен он не идеально, и в нем сгруппированы дела, никак не связанные друг с другом, отсутствует некая информация, ну и прочие недочеты. Существуют разные компьютерные программы для составления списка дел, и они могут исправить эту ситуацию, однако из всех, которые нам встречались, самой амбициозной является *ETM*.

Хотим сообщить для начинающих, что автор немало знает о том, Как Все Успеть [Getting Things Done, GTD] — методике тайм-менеджмента, которая стала популярна в 2002 году благодаря книге с тем же названием.

Согласно GTD, самое главное — записывать информацию о задачах, а не пытаться удержать ее в голове, а затем разбивать задачи на более мелкие, да на них

и концентрироваться. Таким образом вы более четко сосредоточитесь на том, что надо сделать сейчас, вместо того, чтобы прокручивать в голове ужасный бесконечный список неотложных дел.

В *ETM*, который написан на Python и использует Qt для интерфейса, вы можете получить обычный PIM-образный вид входящих задач на текущий день, неделю или месяц. Но вы также можете группировать задачи по папкам или присваивать им тэги (например, «семья» или «бизнес»).

Одна из особо приятных функций — привязка к задачам таймеров: можно запустить таймер, приступив к выполнению задачи, а закончив, остановить его. В итоге вы увидите, сколько времени ушло на выполнение задачи.

«Одна из приятных функций — привязка к задачам таймеров.»



Вы можете настроить повторяющиеся задачи и затем удалить их отдельные экземпляры или все целиком.

ETM имеет собственный синтаксис для описания задач. Например:

* Встреча с персоналом @s пт 2р @e 90 @l Конференц-зал @t встреча

Это означает, что у вас будет встреча в пятницу, в два часа дня, которая займет 90 минут, будет проходить в конференц-зале и имеет тэг «встреча». Первые 10 минут использования программы все это выглядит весьма странно, но вскоре обретает смысл.

Поскольку *ETM* обладает бесконечными возможностями комбинирования задач, напоминаний и примечаний, стоит сначала изучить ее на файлах примеров с сайта проекта (сохраняйте их как `~/etm/data/`).

Сортировщик файлов

Plgen

Версия: 1.4 Сайт: <http://sourceforge.net/p/plgen/>

Сейчас это звучит смешно, но мы помним, сколько эмоций вызвал у нас первый флэш-плеер MP3. В нем было аж 32 МБ памяти, и стоил он 40 фунтов. С тех пор цены на флэш-плееры сильно упали, и сейчас можно найти обычное 4-ГБ устройство всего за десятку. Однако эти супердешевые MP3-плееры имеют существенный недостаток: в них не создать плей-листы. Они просто воспроизводят песни в алфавитном порядке имен файлов, хранящихся на устройстве.

Чтобы решить эту проблему, вам придется самому переименовать MP3-файлы. Например, если вы хотите, чтобы **Foo.mp3** воспроизводилось раньше **Bar.mp3**, вы можете использовать такие имена файлов, как **aaa_Foo.mp3** и **bbb_Bar.mp3**. Однако делать это вручную в огромной музыкальной коллекции — монументальная задача, и для решения этой проблемы на свет появился *Plgen*. Распакуйте файл

.tar.gz, перейдите в полученную директорию и введите

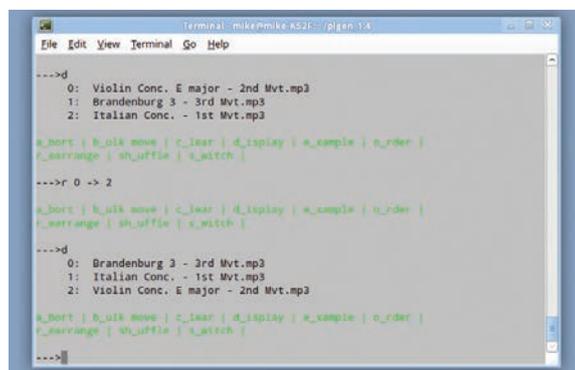
```
python Plgen.py /home/foo/music/
```

Замените `/home/foo/music/` на путь к вашим MP3-файлам, или вашему MP3-плееру, если он подмонтирован. Введя **d**, вы увидите текущий порядок файлов в вашем плей-листе, и у каждого будет порядковый номер. Вы сможете позднее использовать эти номера, чтобы изменить порядок воспроизведения файлов. Например, желая переместить песню, занимающую позицию 1, на позицию 4, вы вводите:

```
r 1 -> 4
```

Снова нажмите на **d**, чтобы увидеть результат. Можно перемещать группы песен в плей-листе с помощью команды **b**

«Дешевые MP3-плееры имеют недостаток: не создать плей-листы.»



Вот пример того, как переместить первую песню в списке на последнее место.

(**bulk** — крупная партия) — введите отдельно **b**, чтобы увидеть, как работает синтаксис. Чтобы плей-лист составлялся в случайном порядке, введите **sh**. А когда закончите, просто введите **q**, для выхода.

Если вы теперь посмотрите на имена файлов, то увидите, что *Plgen* выставил перед ними собственную систему нумерации:

```
__000000__ Foo.mp3
__000001__ Bar.mp3
```

Имена файлов выглядят довольно уродливо, но для очень дешевых MP3-плееров, у которых нет дисплея, это не так страшно. Главное, что порядок теперь определяется этими номерами, то есть по сути вы сумели создать плей-лист.

Утилита поиска файлов

Catfish

Версия: 0.6.3 Сайт: <https://launchpad.net/catfish-search>

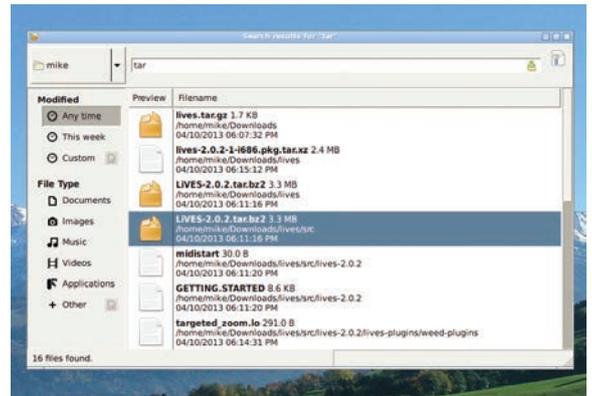
Искать файл в командной строке можно двумя способами: использовать **find** и использовать **locate**. Первая программа осуществляет поиск сразу же — то есть просматривает все указанные директории, смотрит на имена файлов внутри них и ищет совпадения. А вот в случае **locate** имеется предварительно созданная база данных имен файлов, которая сильно ускоряет поиск. Обратной стороной здесь является то, что системе нужно периодически обновлять базу данных **locate**, и она может устареть.

Catfish являет собой графическую оболочку этих двух программ с собственными дополнительными функциями. По умолчанию он включен в Xubuntu, но должен работать на любом дистрибутиве, где есть Python и *PyGtk* (мы тестировали его на Arch Linux). Когда вы его запустите, перед вами появится небольшое диалоговое окно с текстовым полем вверху и панелью результатов внизу. Введите имя

файла (или часть имени), нажмите на Enter — и если обнаружатся совпадения, вы увидите их внизу. *Catfish* быстр и прост в работе, но в этом нет ничего особенно выдающегося.

Однако щелкните по значку настройки вверху справа и затем выберите Show Advanced Settings. Теперь вы можете ограничить поиск файлами определенных типов или файлами, датированными определенными числами. Опция Other в списке File Type позволяет выбирать среди множества типов MIME. С помощью значка настройки также можно задать поиск по всему тексту (что, как и следует ожидать, куда медленнее обычного поиска), и вы также можете просматривать скрытые файлы. Кстати, опция командной

«Вы можете ограничить поиск файлами определенных типов.»



В панели результатов щелкните правой кнопкой, чтобы открыть файл или увидеть его расположение в файловом менеджере.

строки `--thumbnails` заставляет *Catfish* отображать чуть более крупные по размеру значки в качестве результатов, однако, к сожалению, содержимое файлов не отображается.

И все же *Catfish* — солидный и быстрый небольшой инструмент поиска, и это отличное дополнение к легковесной среде рабочего стола. Если вы намерены избегать алчных до памяти Gnome и KDE, но при этом не хотите возиться с оболочкой всякий раз, когда нужно найти файл, он станет отличным решением.

Инструмент стрессового тестирования сайта

Siege

Версия: 3.0.0 Сайт: www.joedog.org/siege-home/

В начале 2000-х существовал забавный интернет-феномен под названием Slashdot Effect. Работал он так: например, вы создавали web-страницу, где показывалось, как сделать в Linux нечто нереально крутое. Ребята из Slashdot узнавали об этом и размещали новость со ссылкой на вашу страницу. И сотни тысяч читателей Slashdot принимались щелкать по ссылке, прорываясь на вашу страницу, что создавало чудовищную нагрузку на полосу пропускания и CPU, и в результате вы могли остаться без web-сервера.

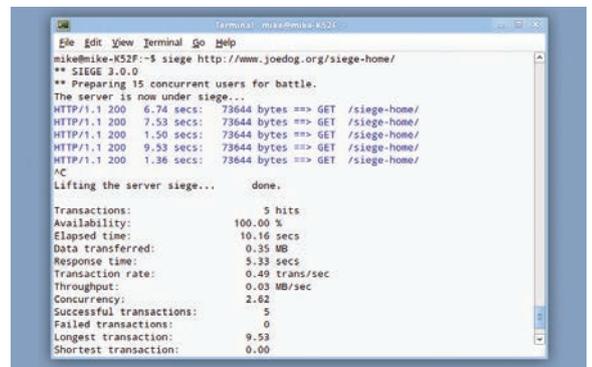
В наши дни такое случается не столь часто, но все-таки перед выпуском в Сеть сайт стоит подвергнуть стресс-тесту. С помощью скриптов *Bash* и **wget** можно соорудить инструмент, который начнет долбить по серверу — однако специально для этой работы уже создан *Siege* [англ. осада].

Будучи инструментом командной строки без каких-либо особых зависимостей

помимо стандартного содержимого любого дистрибутива Linux (*glibc* и т. д.), *Siege* очень прост в компиляции и установке. Однако прежде чем запустить его в первый раз, введите **siege.config** для создания файла **.siegerc** в домашней директории. Потом можете лазить по этому файлу; он очень хорошо задокументирован и снабжен множеством примеров. Но если вы хотите немедленно приступить к тестированию сайта, просто введите нечто вроде `siege http://foo.org/bar.html`

В своей настройке по умолчанию *Siege* готовит «15 одновременно работающих пользователей к битве», то есть имитирует попытки 15 человек одновременно получить доступ к URL снова и снова. Строчки

«Перед выпуском в Сеть сайт стоит подвергнуть стресс-тесту.»



100% доступность — хороший знак: сервер справляется с запросами.

синего цвета показывают, сколько времени сервер тратит на ответ — чтобы установить запросы, нажмите на Ctrl+C. Затем вы увидите сводку, где будет указано общее количество транзакций, скорость передачи данных, доступность сайта (если она менее 100 %, значит, сервер не смог ответить на все запросы), и прочая информация.

Это основные функции *Siege*, но он способен на куда большее: можно имитировать запросы POST, создавать списки URL и заставлять *Siege* выбирать их случайным образом, генерировать лог-файлы и много чего еще. Но с ним надо поосторожнее: если вы укажете ему не тот URL, дело может закончиться DoS-атакой на кого-то...

HotGames Развлекательные приложения

Стрелялка в стиле ретро

Colorful

Версия: 1.1 Сайт: <http://svgames.pl/en/down>

Изначальная версия этой игры была написана за 48 часов для соревнования по онлайн-кодированию. Темой было «ты подлец», что требовало от участников необычного сюжета. В *Colorful* вы — парень, влюбленный в девушку, у которой умерла мать, и, поскольку она погрузилась в депрессию, вы отправляетесь на охоту за цветными штучками, чтобы вернуть ее миру краски.

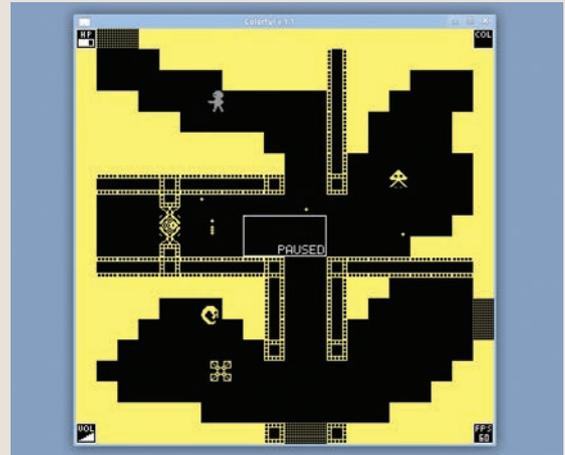
Однако сводится это все к стрелялке с исследованием помещений с простым управлением (используйте стрелки для перемещения, Z для выстрела влево, X для выстрела вправо). Игра написана на Pascal, но если вы извлечете двоичный ZIP-файл, можно скопировать `libbass.so` в `/usr/local/lib` и затем запустить `./colorful`. Исходный код на Pascal предоставлен в другом ZIP, но обратите внимание, что хотя он и открыт, разработчик все же просит всех желающих

выпускать производные версии предварительно согласовывать это с ним.

Игровой процесс прост, но отличается динамичностью и напряженностью. По мере прохождения на вашем пути появляются разные монстры и электронные двери; у вас есть счетчик здоровья, так что пару ударов вы выдержите, но умерев, вам придется вернуться к началу. Подобрал цветную вещь, несите ее своей девушке, которая ждет вас на стартовом экране. Удобно, что предусмотрен режим обучения.

Colorful при самом пылком воображении не назовешь грандиозной игрой, и особого разнообразия здесь нет,

«Охотьтесь за цветными штучками, чтобы вернуть миру краски.»



► Вы — тот серый парень сверху слева, и все вокруг пытается вас убить. Вот так все просто.

однако сама суть игры сработана очень хорошо. Управляющие органы чутки, а крайне аскетичная графика означает, что предметы, которые в вас швыряют ваши враги, не будут затушеваны чересчур сложным фоном. Что, в свою очередь, означает, что за свою смерть вы отвечаете сами. Что, в свою очередь, означает, что вы никогда не будете расстроены, даже в самых сложных комнатах. И это хорошо.

Ролевая игра

Choria

Версия: 0.4.0 Сайт: <http://code.google.com/p/choria/>

Что является самым важным элементом для великолепной RPG? Захватывающий сюжет или проработанная боевая система? Если вы совсем потеряли разум, вы можете решить, что RPG нужны долгие часы «шлифовки» — то есть выполнение снова и снова одних и тех же задач, чтобы заработать опыт. Это кажется глупым, но некоторые фанаты ролевых на самом деле любят подобную «рутину» игры, поскольку она не требует большой концентрации или решения сложных головоломок. Если у вас выдалась свободная полчасика, можете снова и снова мочить монстров и собирать золотишко.

На то у нас и *Choria*. Это «исключительно шлифовка и рутина», по словам разработчика, и прежде чем вы вообще приступите к игре, вам придется пройти через самую главную рутину:

компиляцию исходного кода. По какой-то непонятной причине игра требует движка Irrlicht 3D, несмотря на то, что *Choria* полностью выполнена в 2D с совершенно невыразительной графикой.

Что до игрового процесса, это абсолютно стандартная ролевка: вы начинаете в городке с парой лавочек и возможностью торговли вещами. Выйдите в большой мир, и вас ждут случайные встречи с врагами; убейте их, чтобы получить золото и баллы за опыт. Набрав достаточно баллов, вы получите новый уровень и сможете добавить новые навыки в свою коллекцию.

«Вы прогрессируете, и возникает соблазн исследовать.»



► Похоже на штаны, а описание жутко нудное, но за этим скрывается весьма игральная ролевка.

Теперь, очевидно, вы дивитесь: а зачем рассказывать о такой заурядной игре здесь? На самом деле, она довольно забавная. Мы думали, что проведем часок за шлифовкой, просто чтобы получить новый уровень, но вы очень быстро прогрессируете, и у вас возникает соблазн исследовать карту. Погибнув в битве, вы телепортируетесь назад в родную деревню, но у вас остается ваш уровень, и вам не нужно начинать с самого начала. А если вам надоест основная игра, у вас есть редактор карт и мультиплеер.

Программа просмотра репозитория Git

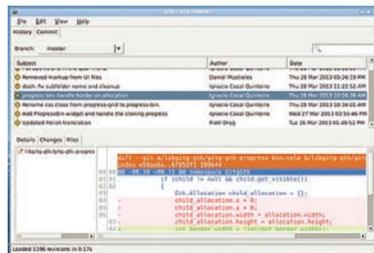
Gitg

Версия: 0.2.7 Сайт: <https://git.gnome.org/browse/gitg/>

Ну как не любить Линуса Торвальдса? Когда система управления исходным кодом (SCM) перестала удовлетворять его потребности, он взял да и написал собственную. С нуля. Многие из нас обошлись бы мерами попроще, пытаясь изменить существующую систему или запустить проект, в расчете на новых разработчиков. Но нет, старый добрый Линус не медля взялся за код и в итоге получил *Git* (который назвал в свою честь, как и ядро, по его собственным словам).

Это было в 2005, и с тех пор *Git* стал невероятно популярной SCM, особенно для крупных проектов с большим количеством участников и подразделений. *Git* не сильно сложен в использовании из командной строки, однако графические инструменты часто помогают упростить его навигацию по проекту и просмотр изменений. *Gitg* служит тому хорошим примером: это основанная на *GTK* программа просмотра репозитория *Git* со списком изменений вверху и подробностями внизу.

Щелкните по изменению на экране, и помимо информации об авторе и исполнителе вы сможете просмотреть диаграммы, показывающие, какие файлы подверглись изменению (и до какой степени). Затем во вкладке Changes можно просмотреть выделенные цветом различия и проследить точные изменения в файлах построчно. Имеется и альтернативная горизонтальная раскладка, которая особенно хорошо смотрится на широких экранах, и всего парой щелчков можно загрузить из сети самую свежую версию репозитория.



▶ Плеснули цвета — и различия в файлах стали гораздо заметнее.

Linux на слое Windows

Atratus

Версия: 0.7 Сайт: <http://atratus.org>

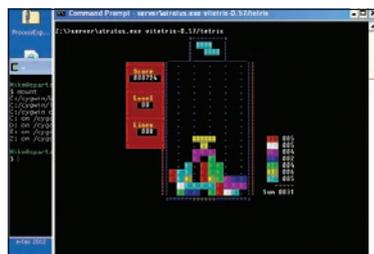
Что?! О программе Windows рассказывают в HotPicks? Да, это в некотором роде против правил, но код ее открыт, и она очень тесно связана с Linux. *Atratus* — это уровень двоичной совместимости, способный запускать бинарники Linux на системах Windows. Вылитый *Wine*, только работает по-другому.

Говоря «работает», мы на самом деле подразумеваем очень небольшое подмножество программ Linux. Не обольщайтесь, что вы сможете запустить в Windows такого монстра, например, Linux-версию *LibreOffice* (да оно и незачем, поскольку она и так многоплатформенная). Но ряд небольших программ командной строки весьма нормально работают, в том числе и некоторые игры, и утилиты *BusyBox*.

Запуск их в Windows достаточно прост: распакуйте ZIP-файл и дважды щелкните по файлам BAT для доступа к программам в образовавшейся директории. Пока *Atratus* работает только на 32-битной версии

Windows, и хорошо протестирован только на Windows XP. Он зависит даже от недокументированных функций в Windows API и вполне может сломаться, когда Microsoft выпустит обновление.

Но тем не менее он интересен. Большинство хороших программ с открытым кодом так или иначе портированы в Windows, однако может оказаться, что как раз то симпатичное приложеньце, которое вы используете, осталось за бортом. Впереди еще большой путь, но мы будем следить за развитием проекта. **LXF**



▶ С учетом явной ограниченности, место ему скорее на стадии 0.2, а не 0.7, но все же впечатляет.

Также вышли

Новые и обновленные программы, тоже достойные внимания...

▶ Dianara 0.3

Клиент для распределенной социальной сети Diaspora.
<http://dianara.nongnu.org>

▶ pwgrep 0.8.3

Менеджер паролей на базе *Bash/Awk*, использующий для шифрования *GnuPG*.
<http://pwgrep.buetow.org>

▶ CorneliOS 3.4r10

Впечатляющая «среда рабочего стола», которая работает внутри web-браузера.
www.cornelios.org



▶ Выглядит, работает и даже пахнет, как полноценная ОС — но это web-приложение.

▶ ximp3 0.1.18

MP3-плеер командной строки, основанный на декодере *Xing*.
<http://freecode.com/projects/ximp3>

▶ Marble 1.5

Очень красивый виртуальный глобус и атлас мира для KDE.
<http://marble.kde.org/>

▶ Podget 0.6.14

Сборщик подкастов, программируемый (например, в *Cron*) на определенное время срабатывания.
<http://podget.sourceforge.net>

▶ Eko 0.8.1

Простой и легковесный звуковой редактор.
<http://semiletov.org/eko/>

▶ O. H. R. RPG.C.E 2013-04-10

Создавайте ролевки наподобие *Final Fantasy*.
<http://rpg.hamsterrepublic.com/ohrrpgce/>



▶ Оторвитесь, как в 1992-м, и создайте ролевку своей мечты.

На диске

Дистрибутивы, приложения, игры, подкасты и многое другое...

Лучшее из Интернета, упакованное в тугие 8 ГБ классного DVD.



Дистрибутивы

Дистрибутивы Ubuntu Remix — любопытная штука. Внутри разница между ними не так уж велика: в конце концов, все они созданы на более или менее одном исходном коде. Но возьмем, например, ДНК, которая есть не что иное, как исходный код для всех живых существ: разница между человеком и шимпанзе — всего 1%. Так что мелочи могут приводить к колоссальным различиям...

Чтобы помочь людям понять разницу между дистрибутивами, мы создали искатель дистрибутивов от Tux-Radar на www.tuxradar.com/content/distro-picker-0. Это web-приложение, помогающее выбрать дистрибутив именно для вас. Естественно, его совет стоит воспринимать с изрядной долей сомнения, но его истинная цель вовсе не навязать вам какую-то версию Linux, а помочь осмыслить, что именно важно для вас в дистрибутиве (а заодно и немного развлечь). В конце концов, создание дистрибутивов — это умение идти на компромисс. Нельзя получить блестящую графику и отличную скорость на старых компьютерах; или самые новые программы — и стабильность. Хитрость поиска нужного дистрибутива в том, чтобы найти такой, чьи создатели ценят те же компромиссы, что и вы сами, потому что иногда даже 99% совпадения бывает мало.

Ben

» Важно ВНИМАНИЕ!

Порченные диски

В маловероятном случае какого-то дефекта вашего LXF DVD обращайтесь, пожалуйста, по адресу disks@linuxformat.ru или телефону +7 (812) 309-0686.

Дистрибутивы Linux

Ubuntu Remix



Ни один дистрибутив не вызывает столько споров, сколько Ubuntu. Похоже, люди его или любят, или ненавидят — третьего не дано. Вероятно, самой одиозной функцией является включение результатов поиска Amazon в панель Dash — функция, которую Ричард Столлмен обозвал «шпионским ПО».

Мы понимаем важность конфиденциальности, но по-прежнему считаем Ubuntu отличным дистрибутивом Linux — и создали собственный ремикс Ubuntu, где поиск Amazon по умолчанию убран из линз. Так что вы сможете наслаждаться своим Unity, не переживая насчет того, что компьютер рассылает ваши данные по всей Сети.

Дистрибутивы Linux

Варианты: Kubuntu & Xubuntu

Unity — не для всех. Народ в Canonical это понимает, и они выпустили серию официальных версий с другими средами рабочего стола. Kubuntu, вероятно, самая популярная из них. Она больше не разрабатывается Canonical, но те по-прежнему предоставляют для нее ресурсы и репозитории, и, понятное дело, базу, на которой она создана.

KDE, среда рабочего стола, на которой создан Kubuntu, в высшей степени настраиваема. Практически каждый ее аспект можно изменить в соответствии с вашими потребностями.

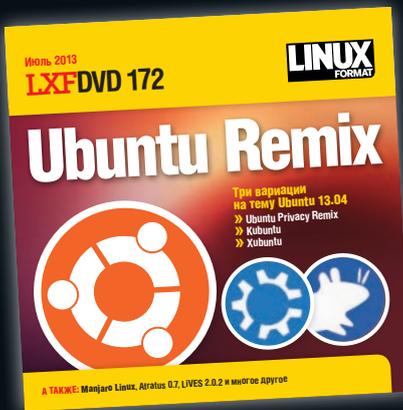
Рабочий стол попроче

Конечно, не все жаждут уметь настраивать каждую мелочь. Некоторым нужна несложная среда, которая

просто работает. Для них мы включили Xubuntu на базе Xfce, которая не лезет под ноги, просто позволяя вам делать свою работу на компьютере.



» Kubuntu: теперь чуток подбавил пурпура.



Новичок в Linux? Начните здесь!

- » Что такое Linux? Как его установить?
- » Есть ли в нем эквивалент MS Office?
- » Зачем нужна командная строка?
- » Как устанавливать программы?

Ответы есть в [Index.html](#) на диске.

Из журнала

Manjaro

Пробуйте самую свежую версию этого дистрибутива на базе Arch (обзор на стр. 19) в режиме live с диска.

Ответы

Все ваши проблемы с открытым кодом решаются именно здесь.



LXFHotPicks

Atratus

Запускайте программы Linux на Windows безо всяких эмуляторов.

Catfish

GUI для команд find and locate (поиска и определения местоположения) немного упрощает их использование.

Choria

2D-MMORPG — только рутинные задачи. Но на самом деле это куда интереснее, чем кажется. Правда!

Colorful

Поиск цветных артефактов, дабы снова сделать жизнерадостным мир вашей погруженной в депрессию девушки — вот цель этой абстрактной аркады.

ETM

Менеджер задач на базе методики Как Все Успеть [Getting Things Done].

Gitg

Сделайте Git чуть дружелюбнее к пользователю с помощью этой программы просмотра на GTK.

G4L

Инструмент создания образов разделов диска.

LIVES

Один из лучших видеоредакторов в Linux.

Plgen

Создавайте плей-листы в плей-листах MP3, изначально этого не предусматривавших.

Siege

Инструмент стрессового тестирования сайтов в процессе их разработки.

Sxiv

Программа просмотра изображений, управляемая с клавиатуры командной строкой.



» LIVES значительно упростит вам процесс создания видео про котят.

И еще!

Системные инструменты

Главное

Checkinstall Установка tarball с помощью менеджера пакетов.

GNU Core Utils Основные утилиты, которые обязаны быть в каждой ОС.

Hardinfo Средство тестирования системы.

Plop Простой менеджер загрузки для запуска операционных систем.

RaWrite Создавайте загрузочные диски в Windows.

SBM Независимый от ОС менеджер загрузки с несложным интерфейсом.

Wvdial Соединяйтесь с Интернетом через модем dial-up.

Чтение

Книжная полка

The Cathedral and the Bazaar [Собор и Базар] Классический текст Эрика С. Реймонда [Eric S Raymond] о преимуществах открытой разработки.

Linux Kernel in a Nutshell [Ядро Linux в двух словах] Введение в ядро от Великого магистра Грега Кроа-Хартмана [Greg Kroah-Hartman].

Справочник администратора Debian Руководство для сисадминов.

Словарь Linux Linux от А до Z.

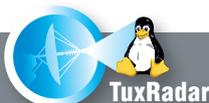
Окунитесь в Python Мастер-класс по этому популярному языку.

Руководство по созданию скриптов Bourne Shell Осваивайте скрипты оболочки.



» Гуру ядра и писатель Грег Кроа-Хартман.

Подкасты



S5 E7 Ядерный Сондерс

Новости [News] Ветеран LXF Майк Сондерс [Mike Saunders] возвращается в единственную серию.

Открытое голосование [Open Ballot] А вы бы стали платить за Linux?

Открытия недели [Discoveries of the week] Retro Arch, Into The Dead, Terminator, IRC.

S5 E7 Султаны свинга

Новости [News] Google Summer of Code и Linux on Alienware.

Открытое голосование [Open Ballot] Понимает ли Microsoft юмор?

Открытия недели [Discoveries of the week] Fastogenc, Adblock Plus, боль в глазах, Gmail вредит.



Пропустили номер?



Закажите его через «Линуксцентр» по адресу www.linuxcenter.ru! Журналы доставляются и в печатной, и в электронной форме, так что с момента открытия браузера до получения нужного вам выпуска LXF может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

LXF169 Апрель 2013



250 руб.

- » **Linux — что это?** Познавательная вивисекция
- » **ОС для Raspberry Pi** Дать дистрибьютерам компьютеру-малютке
- » **WebRTC** И браузер с браузером говорит... в реальном времени
- » **Лейся, песня** Управляем фонотекой из командной строки

Печатная версия: shop.linuxformat.ru/lxf_169/
PDF-версия: shop.linuxformat.ru/elxf_169/

LXFDVD: Fedora 18 в версии Linux Format, 5 дистрибутивов для RPi и еще 3 дистрибутива, 10 книг о Linux (на английском языке), горячие новинки и прочее...

LXF170 Май 2013



250 руб.

- » **Взлом Сети** Учимся парировать атаки
- » **Дистрибутивы для продвинутых** Характер нордический
- » **Samba 4** Докладываем: есть совместимость с Active Directory!
- » **Kazan** Рабочий стол как герой кинофильма

Печатная версия: shop.linuxformat.ru/lxf_170/
PDF-версия: shop.linuxformat.ru/elxf_170/

LXFDVD: PCLinuxOS, Porteus 2.0 и еще 3 дистрибутива, 10 книг о Linux (на английском языке), горячие новинки и прочее...

LXF171 Июнь 2013



250 руб.

- » **Звездные полсотни** Дистрибутивы Linux на любой вкус
- » **Инфографика для всех** Выбираем достойный инструмент
- » **Secure Boot** Защита или приманка в капкане?
- » **Web-приложения «под ключ»** Развернем за пару щелчков

Печатная версия: shop.linuxformat.ru/lxf_171/
PDF-версия: shop.linuxformat.ru/elxf_171/

LXFDVD: OpenSUSE 12.3, Trisquel и игровой дистрибутив LXF, 10 книг о Linux (на английском языке), лекция Ричарда Столлмена, горячие новинки и прочее...

А чтобы не упустить ничего, оформите подписку! Все, кто подписался на печатную версию журнала через www.linuxformat.ru/subscribe или www.linuxcenter.ru, получают электронную версию в подарок! Через shop.linuxformat.ru вы также сможете приобрести предыдущие выпуски LXF (если успеете).

Подписывайтесь на www.linuxformat.ru/subscribe/

Телефоны отдела подписки:

- » Санкт-Петербург (812) 309-0686
- » Москва (499) 271-4954



Linux Format ВКонтакте

Вступайте
в нашу группу
vk.com/linuxform

На странице
журнала Linux Format
ВКонтакте вы найдете:

- » Множество новостей о Linux
- » Статьи из архива LXF
- » Анонс свежего выпуска LXF и часть статей из него
- » Живое общение и онлайн-консультацию по подписке на наше издание

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: MDM 1.2, менеджер драйверов MiniDrivers и многое другое

Linux Mint 15
Ядро 3.8 » Gimpom 1.8 » МАТЕ 1.6 » База пакетов Ubuntu 13.04
32- и 64-разрядная сборки



А ТАКЖЕ: Manjaro Linux, Atlantis 0.7, LIVES 2.0.2 и многое другое



Три вариации на тему Ubuntu 13.04
» Ubuntu Privacy Remix
» Kubuntu
» Xubuntu

Ubuntu Remix

Июль 2013
LXF DVD 172

LINUX
FORMAT

Июль 2013
LXF DVD 172

LINUX
FORMAT

Содержание

Сторона 1

ДИСТРИБУТИВЫ

Ubuntu Privacy Remix Сборка Ubuntu 13.04, где исключен поиск на Amazon по умолчанию, 32-разрядная сборка (загрузка с LXFEVD)
Kubuntu Ремикс Ubuntu 13.04 с рабочим столом KDE, 32-разрядная сборка (загрузка с LXFEVD)
Xubuntu Ремикс Ubuntu 13.04 с рабочим столом Xfce, 32-разрядная сборка (загрузка с LXFEVD)
Manjaro 0.8.5 Дистрибутив на базе Arch с более дружественным интерфейсом администрирования, 32-разрядная сборка (загрузка с LXFEVD и ISO-образ)

Сторона 2

ДИСТРИБУТИВЫ

Linux Mint 15 Установочные DVD (ISO-образы)
 » **С рабочим столом Cinnamon**
 32- и 64-разрядные сборки
 » **С рабочим столом MATE**
 32- и 64-разрядные сборки

ДОКУМЕНТАЦИЯ — 10 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 Раймонда [Eric S Raymond] «Сбори и базар»
The Debian Administrator's Handbook Руководство администратора, написанное разработчиками Debian

Dive Into Python Учебник по программированию на Python
Intro to Linux Начальное руководство по Linux
Linux Dictionary Словарь Linux, объясняющий специфическую терминологию
Linux Kernel in a Nutshell Описание ядра Linux, созданное одним из его выдающихся разработчиков — Грегом Кроа-Хартманом [Greg Kroah-Hartman]
System Administrator's Guide Руководство по базовому администрированию Linux
GNU Tools Summary Руководство по работе в командной строке и обзор основных утилит GNU

NOTPICKS
Atralus 0.7 Среда для запуска программ Linux на Windows
Calfish 0.6.3 Утилита поиска файлов
Cloria 0.4.0 Игра, 2D MMO RPG
Colorful 1.1 Игра, стрелялка в стиле retro
ETW 2.1.3 Менеджер задач, основанный на методике Как Все Успеть [betting Things Done]
Gig 0.27 Программа для просмотра репозитория Git
G4L 0.44 Утилита клонирования разделов диска
LIVES 2.0.2 Один из лучших видеоредакторов в Linux
Pigen 1.4 Сортировщик файлов
Siege 3.0.0 Инструмент для стрессового тестирования сайтов в процессе разработки
Skiv 1.1 Программа просмотра изображений, управляемая с клавиатуры

Пожалуйста, перестаньте использовать именем Адного Диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru
ДЕФЕКТИВНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данной 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 новым 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, предоставленных нашей программой или данных. Прежде чем устанавливать какое-либо ПО на компьютер, пожалуйста, скачайте программу для загрузки с ее официального сайта.
 Тираж изготовлен ООО «Уральский электронный завод», 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПР ВАР 77-15.

Создание установочных дисков при помощи cdcrcord

Самый быстрый способ записать ISO-образ на чистую матрицу — это обратиться к программе *cdrecord*. Для всех перечисленных ниже действий потребуются права суперпользователя-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 (скорее всего, в вашей системе присутствует только 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
```

Для записи ISO-образа вам осталось набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не принадлежите к любителям командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее от имени 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?

Использовать Linux для записи компакт-диска не обязательно. Программы вроде *cdrecord* просто переносят двоичные данные на чистую матрицу. Все необходимые файлы уже включены в ISO-образ, который распознается любой операционной системой, будь то Linux, Windows, Mac OS X или AmigaOS.

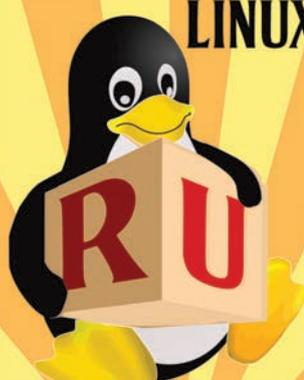
Нет устройства для записи дисков?

Если у вас нет устройства, с помощью которого можно было бы записать образ на диск, можно найти какого-нибудь друга или организацию, у кого есть компьютер с дисководом, и прожечь диск у них. Опять-таки, вам 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.

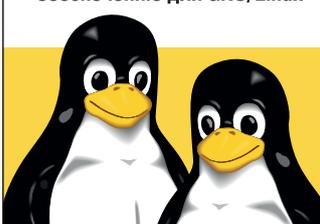
ФОРУМ №1

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

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/

V МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

ИНФОФОРУМ ШАНХАЙ

ДОВЕРИЕ И БЕЗОПАСНОСТЬ В ИНФОРМАЦИОННОМ ОБЩЕСТВЕ

Российская неделя КИБЕР-БЕЗОПАСНОСТИ

23-28 в КИТАЕ СЕНТЯБРЯ

Участник **SIBT-2013**



CHINA.INFOFORUM.RU

LINUX FORMAT

Главное в мире Linux

Как разместить рекламу в разделе Classifieds?

1/4 полоса (210 × 297 мм)	165 200 руб.
1/2 полосы горизонтально (197 × 144 мм)	88 500 руб.
1/2 полосы вертикально (102 × 278)	88 500 руб.
1/4 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

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
Расчитан на 250 ПК



65 240 руб.

DR.WEB OFFICE SHIELD NEO
Расчитан на 50 ПК

Dr.WEB®

Linux center
www.linuxcenter.ru

WWW.LINUXCENTER.RU/SHOP/ANTIVIR/DR_WEB/

UnixEducation Center
Россия, 190000, Санкт-Петербург
Черноморский переулок, дом 4
Тел.: + 7 (812) 611-1575






RHCSA / RHCE Certification



PEARSON
VUE
AUTHORIZED
TEST CENTER



PROMETRIC
TEST CENTER

Футболки GNU/Linux

БОЛЕЕ 100 ВИДОВ. ВСЕ РАЗМЕРЫ И ЦВЕТА



WWW.LINUXCENTER.RU/SHOP/GIFTS/



В августовском номере

На свободу!

Выцарапайте свои данные из когтей Google и иже с ним, и смейтесь над страшилками про конфиденциальность. Свободное ПО дает нам для этого средства — ну так пользуйтесь!

Производные Ubuntu

Берите от Ubuntu лучшее, а худшего нам не надо! Рассматриваем лучшие ремиксы *buntu.

Внутри EFF

Люди из фонда Electronic Frontier борются за ваше цифровое будущее. И вот каким образом...

Чего ждать от KDE

Сунем нос в процесс создания рабочего стола для любителей понастраивать и разберемся, почему KDE 5 не бывать.

Содержание будущих выпусков может меняться — вдруг мы будем уезжать от гуглопоклонников...

LINUX FORMAT

Главное в мире Linux

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, Ирина Шулакова

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

Издатель

ЗАО «Мезон.Ру»

Отпечатано в ООО «Ланинь»

188330, Ленинградская обл., Гатчинский р-н, пос. Сиверский, Вокзальная ул., 4
Заказ 8002

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ

Редактор Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.com

Заместитель редактора Эндрю Грегори [Andrew Gregory]

andrew.gregory@futurenet.com

Редактор диска Бен Эверард [Ben Everard] ben_everard@futurenet.com

Художественный редактор Эфраин Эрнандес-Мендоса

[Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.com

Выпускающие редакторы Андреа Болл [Andrea Ball], Рой Дилэйни [Roy Delaney], Стивен Рейнс [Steven Raynes]

ПОДГОТОВКА МАТЕРИАЛОВ

Джон Бэкон [John Bacon], Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Алекс Кокс [Alex Cox], Бен Эверард [Ben Everard], Марко Фиоретти [Marco Fioretti], Дэвид Хейвард [David Heyward], Сюзан Линтон [Susan Linton], Грэм Моррисон [Graham Morrison], Джонатан Робертс [Jonathan Roberts], Майк Сондерс [Mike Saunders], Евгений Балдин, Андрей Грозин, Артем Зорин, Андрей Сухарев, Александр Толстой, Александр Хрюкин, Алексей Федорчук

Художественные ассистенты

Стейси Блэк [Stacey Black], Анна Мак-Ги [Anna McGee]

Иллюстрации Шейн Коллиндж [Shane Collinge], Ely Walton Illustrations, iStockPhoto

КОНТАКТНАЯ ИНФОРМАЦИЯ

UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW

Тел. +44 01225 442244, email: linuxformat@futurenet.com

РОССИЯ:

Санкт-Петербург (редакция):

Лиговский пр., 50, корп. 15

Тел. +7 (812) 309-0686

Представительство в Москве:

Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)

Тел./факс +7 (499) 271-4954

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 исключительное 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 диски — CD или DVD, 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 повреждения 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.futurepic.com>



© Linux Format 2005

© Future Publishing Ltd 2005

BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO

16+



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



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