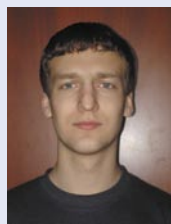


## Колонка главного редактора



Начало осени внесло свой традиционный колорит: корпоративный мир постепенно отходит от летней спячки, а особые активисты и вовсе погружаются в рабочую атмосферу с головой. Возвращению людей к реальности в первую очередь способствуют СМИ, бросающие все свои силы на попытки шокировать свежими деловыми потрясениями. К бою готова и наша редакция, посвятившая значительную часть этого выпуска некоторым подробностям последних событий мира FOSS.

Тем временем в Москве подошел к концу сезон крупных IT-выставок, украсив сентябрь мероприятиями, в рамках которых были представлены второй LinuxWorld Russia и второй же LinuxLand. Несмотря на во многом схожий состав участников, не могу не подчеркнуть ощущения значительно более приятной и дружелюбной атмосферы (с позиций обычного обывателя) в последнем случае.

Вероятно, во многом виновата та показная «ориентация на крупный бизнес», что была заявлена организаторами LinuxWorld. И судя по тому, с каким остервенением выражал свое неудовольствие царящей вокруг коммерциализацией Open Source один из посетителей стенда нашего журнала, прочувствовал это не один я. Не стоит делать из FOSS нечто недоступное тем самым энтузиастам, что создают его.

Главный редактор  
Дмитрий Шурупов  
(osa@samag.ru)

### «Open Source»

электронное приложение к журналу  
«Системный администратор»  
№12, 3 октября 2006 г.

#### РЕДАКЦИЯ

Исполнительный директор

Владимир Положевец

Главный редактор

Дмитрий Шурупов

Верстка и оформление

Владимир Лукин

Сайт электронного приложения:

<http://osa.samag.ru>

За содержание статьи ответственность несет автор. Все права на опубликованные материалы защищены.

# Новости мира Open Source

## 2X открыла исходный код TerminalServer для GNU/Linux

Компания 2X объявила о выпуске 2X TerminalServer for Linux – терминального сервера с открытым кодом для операционной системы GNU/Linux.

TerminalServer для Linux позволяет пользователям запускать Linux/Windows-приложения на удаленных машинах через любое (даже достаточно медлительное) подключение.

«Если Linux собирается укрепляться на десктопах, системе потребуется терминальный сервер вроде 2X Terminal Server for Linux. Только более продвинутое решение на базе тонких клиентов позволят Linux превзойти «толстые» клиенты Windows в сетях компаний. 2X гордится возможностью внести свой вклад, открыв исходный код терминального сервера для Linux», – прокомментировал исполнительный директор 2X Николаос Макрис (Nikolaos Makris).

2X TerminalServer for Linux использует технологию NX от NoMachine, распространяется свободно (под лицензией GNU GPL) и бесплатно. Подробности о проекте доступны на <http://www.2x.com/news/linuxterminalserver.htm>.

## Организационные изменения в проекте NetBSD

Реорганизация проекта свободной операционной системы NetBSD, что началась еще в далеком 2002 году, формально завершилась в первый день сентября.

Дело в том, что в 90-х были утеряны договоренности, подписанные многими разработчиками с организацией NetBSD Foundation, в рамках которых они участвовали в проекте. И с тех пор некоторые из них продолжали помогать проекту на протяжении многих лет без таких договоров.

В NetBSD Foundation были предприняты попытки связаться с этими разработчиками, чтобы те подтвердили свое намерение продолжать принимать участие в проекте. Но несмотря на все усилия, некоторые так и не откликнулись.

Тогда руководящий комитет NetBSD по членству (Board and Executive Committee on Membership) постановил произвести упразднение доступа для разработчиков без договоров 1 сентября 2006 года. Все были предупреждены за неделю до события.

1 сентября проект NetBSD попрощался с некоторыми своими участниками, которые внесли значительный вклад в его раз-

витие за минувшие 13 лет, поблагодарил всех их и напомнил, что, если время и обстоятельства позволят, вероятно, мы сможем вновь увидеть их лица, что дополнят список из 346 человек, помогающих совершенствовать NetBSD сегодня.

## Вышел GNOME 2.16; планы по GNOME 2.17, 2.18

Начало сентября принесло выход очередной стабильной версии популярной графической среды GNOME – 2.16.

Среди изменений в GNOME 2.16 отмечается появление нового редактора меню Alacarte, утилиты для создания заметок Tomboy, анализатора использования дискового пространства Baobab, экранного диктора Orca, улучшения в медиаплеере Totem, веб-браузере Epiphany и других компонентах среды.

Кроме того, был обнародован приблизительный план релизов следующей ветки для разработчиков популярной свободной графической среды GNOME – 2.17.x.

Первый релиз GNOME 2.17 для разработчиков (2.17.1 Development Release) ожидается 18 октября 2006 года, второй – 8 ноября, третий – 6 декабря, четвертый – 20 декабря, пятый – 10 января 2007 года.

GNOME 2.18.0 Beta 1 (2.17.90), что станет результатом разработок ветки 2.17.x, выйдет 24 января, Beta 2 – 14 февраля, GNOME 2.18.0 Release Candidate (2.17.92) – 28 февраля, а GNOME 2.18.0 Final появится 14 марта.

## FreeDOS 1.0 – свободная реализация DOS готова

Проект по созданию совместимой с DOS свободной операционной системы FreeDOS объявил о завершении работы над первым финальным стабильным релизом – 1.0. Авторы трудились над этим продуктом около 12 лет – начало процесса разработки приходится на 1994 год.

FreeDOS 1.0 может загружаться на одной машине с операционными системами Windows 95-2003/ME/NT/XP, работать с файловой системой FAT32 и большими жесткими дисками (LBA), поддерживает LFN и дисковый кэш (LBACACHE).

В состав свободной ОС входит ПО для управления памятью (HIMEM, EMM386, UMBPCI), альтернатива SHSUCDX и драйвер для CD-ROM (XCDROM), драйвер для мыши с колесом прокрутки (CUTEMOUSE), FDAPM для управления питанием ПК, драйвер UDMA (XDMA, до 4 дисков), такие пользовательские приложения, как мультимедийный проигры-

ватель MPXPLAY (mp3/ogg/wmv и другие форматы), архиваторы (7ZIP, INFO-ZIP zip&unzip), многооконные текстовые редакторы (EDIT/SETEDIT), просмотрщик файлов помощи (HTMLHELP), текстовый просмотрщик (PG).

Распространяется FreeDOS под свободной лицензией GNU GPL. Информация о проекте доступна на <http://www.freedos.org>.

### Погиб Роберт Левин, основатель IRC-сети FreeNode

16 сентября в больнице скончался Роберт Левин (Robert Levin), известный в IRC как «lilo». Роб основал IRC-сеть FreeNode, что является интернет-домом для общения авторов, разработчиков и пользователей проектов программного обеспечения с открытым кодом, и возглавлял ее родительскую организацию PDPC (Peer-Directed Projects Center).

12 сентября Роберт Левин попал в автомобильную аварию во время езды на велосипеде и, получив травмы головы, был отправлен в больницу, где умер 16 сентября.

Сообщество FOSS скорбит по этой утрате. Некоторые называют Роберта «невоспетым героем Open Source».

### Mandriva Linux 2007, Corporate Server 4.0

Компания Mandriva объявила о выпуске новых версий своих дистрибутивов: Corporate Server 4.0 и Mandriva Linux 2007.

Среди новшеств в Corporate Server 4.0 авторы выделяют поддержку технологий виртуализации VMware, Xen и OpenVZ, а также появление утилиты FIBRIC, помогающей устанавливать и настраивать пакеты через веб-интерфейс. Из паке-

тов в CS 4.0: Linux-ядро 2.6.12, MySQL 5.0, PostgreSQL 8.1, Apache 2.2, Samba 3.0.22.

В Mandriva Linux 2007 главным новшеством стало добавление обеих современных технологий, кардинально преобразующих пользовательский рабочий стол, — AIGLX и Xgl. Причем сообщается, что предусмотрена поддержка широкого спектра железа и включила в поставку дистрибутива специальную утилиту для автоматического определения лучшего 3D-решения (AIGLX или Xgl) для текущего аппаратного обеспечения. Из пакетов в Mandriva Linux 2007: Linux-ядро 2.6.17, GCC 4.1 и glibc 2.4, GNOME 2.16 и KDE 3.5.4, Xen 3.0, Firefox 1.5.0.7, OpenOffice.org 2.0.3.

### Novell готовит Suse Linux Enterprise Real-Time

В октябре Novell пополнит линейку корпоративных дистрибутивов, выпустив ОС реального времени — Suse Linux Enterprise Real-Time.

SLERT призвана реагировать на события с минимальными временными затратами. Как сообщил Джастин Штейнман (Justin Steinman), директор по маркетингу решений Novell на базе Linux и открытых платформ, анонс нового продукта состоится 9 октября на Gartner Symposium/ITxpo. Над Suse Linux Enterprise Real-Time работали специалисты из компании Concurrent Computer.

Известен первый крупный клиент: Siemens Medical Solutions воспользуется SLERT для оснащения продуктов магнитно-резонансной интроскопии Magnetom.

### Red Hat — в списке самых быстрорастущих компаний

Американская Linux-компания Red Hat после трех лет впечатляющего роста заняла 29-ю строчку в списке «100 самых

быстрорастущих компаний», опубликованном в журнале Fortune.

Рейтинг «100 Fastest Growing Companies» был создан на основе исследования доходов различных компаний за последние 3 года. За этот промежуток времени Red Hat удалось увеличить свои доходы на 47 процентов. В минувшем году доходы Red Hat составили 278,3 млн. долларов, а чистая прибыль — 79,7 млн. долларов.

### Dunc-Tank заплатит разработчикам Debian

Группа разработчиков, называющая себя Dunc-Tank, объявила о готовности оказать материальную поддержку участникам проекта Debian GNU/Linux.

Dunc-Tank намеревается способствовать тому, чтобы следующая стабильная версия Debian вышла в соответствии с запланированным графиком. В своем анонсе Dunc-Tank охарактеризовала себя «экспериментом», а участников — «независимой группой разработчиков, пользователей и желающих поддержать Debian». Как позже выяснилось, эту «независимую группу» составляют небезызвестные в сообществе люди, давно участвующие в проекте Debian.

В первую очередь Dunc-Tank хочет оплатить работу двух менеджеров релизов Debian: Стива Лангасека (Steve Langasek) наймут на полный рабочий день в октябре, а Андреаса Барта (Andreas Barth) — в ноябре.

Инициатива вызвала полемику в сообществе в связи с тем, что некоторые сочли плохой идеей так «приправлять» энтузиазм финансовыми ресурсами.

Дмитрий Шурупов,  
по материалам [www.nixp.ru](http://www.nixp.ru)  
([osa@samag.ru](mailto:osa@samag.ru))

## Мнение Linux-разработчиков о GPLv3

### Тема Linux и GPLv3 не теряет актуальности

Готовящаяся к выходу новая версия самой популярной лицензии на свободное программное обеспечение не перестает быть актуальной темой для дискуссий в Open Source-сообществе. Впервые мы познакомили с ней наших читателей в статье «GNU GPLv3 на подходе» из второго выпуска «Open Source». Вскоре после публикации первой черновой версии GPLv3 ее стали обсуждать «на высшем уровне» как возможный вариант для самого успешного продукта мира FOSS, о чем вы можете подробнее почитать в статье «Linux-ядро и GPLv3: первые разговоры» из «Open Source» 005.

Теперь же, спустя почти два месяца с момента публикации второй черновой версии GPLv3, появился актуальный повод вернуться к теме Linux и новой лицензии — разработчики ядра сформулировали и представили общественности свою позицию по вопросам GPLv3.

15 сентября группой разработчиков Linux-ядра, в которую вошли James E.J. Bottomley, Mauro Carvalho Chehab, Thomas Gleixner, Christoph Hellwig, Dave Jones, Greg Kroah-Hartman, Tony Luck, Andrew Morton, Trond Myklebust и David Woodhouse, было составлено официальное «положение об отношении к третьей версии GNU General Public License (в виде ее второго чернового варианта), выпущенное некоторыми разработчиками Linux-ядра, выразившими свое мнение исключительно от лица людей, занимающихся поддержкой ядра» — <http://lwn.net/Articles/200422>. Этот документ состоит из шести пунктов — в данной статье все они изложены в виде кратких вольных авторских интерпретаций, отображающих суть.

## Пункт 1. Linux и GPLv2

GNU/Linux смогла доказать, что является самой успешной операционной системой с открытым кодом за всю историю при условии, что не была в этом первой и далеко не одинока в своей открытости на сегодняшний день. Вокруг нее сформировалось огромное сообщество людей, беспрестанно развивающих и совершенствующих код. Во многом Linux удалось добиться этого благодаря GPLv2. Вторая версия лицензии GPL многие годы служила проекту верой и правдой, способствовала ядру достичь нынешних высот и помогает расти дальше. За все это время в ней не было найдено каких-либо значительных проблем, она не нуждается в поправках.

## Пункт 2. Linux, ядро и мир Open Source

Linux, которую устанавливает конечный пользователь, — это не только ядро, но и набор множества дополнительных компонентов, собранных в готовый дистрибутив. Разработчикам таких дистрибутивов, объединяющим различное программное обеспечение в единое целое, тестирующим результат и доводящим его до стабильной формы, приходится заботиться и об отсутствии лицензионных конфликтов. Ведь у всякого приложения может быть лицензия, противоречащая условиям распространения других компонентов системы. Чем менее однородными становятся лицензии у ПО, составляющего дистрибутив, тем больше головной боли возникает у его авторов.

GPLv2 смогла существенно продвигаться в направлении к тому, чтобы стать доминирующей лицензией. Такое доминирование позволяет упростить жизнь разработчикам дистрибутивов: ведь они могут собрать свою систему полностью из компонентов, лицензированных под GPLv2, и забыть обо всех потенциальных проблемах, связанных с лицензиями на отдельные части продукта. Любое обновление GPLv2 (вследствие массового перехода проектов с нее на новую версию) может повлечь к нарушению экосистемы с гегемонией единой лицензией.

## Пункт 3. Linux и свобода

Одна из принципиальных причин успеха Linux — широта и разнообразие сообщества разработчиков и пользователей, без которых немислим нынешний уровень инноваций. При этом у людей с совершенно разными устремлениями (ведь за многими из них стоят коммерческие структуры) получается добиваться совместного решения общих задач во благо друг дру-

гу. Все это возможно благодаря свободе, которая обусловлена лицензией GPLv2, — свободе от навязывания того, как должны и как будут в конечном итоге использоваться разработки проекта. Недопустимо внесение в GPL любых поправок, которые могут поставить под сомнение эту основополагающую свободу.

## Пункт 4. Центральная роль FSF

Проекты, контролируемые организацией Free Software Foundation, составляют важную часть любого Linux-дистрибутива. Однако в FSF все устроено несколько не так, как это свойственно проекту Linux: у них требуют передачи авторского права от каждого из тех тысяч людей, что внесли свой вклад в кодовую базу. Всякий такой «вклад» передается FSF на условии, что будет лицензирован под версиями GPL, которые «будут похожи по духу на ее текущую версию». Соответственно, такая договоренность была принята и разработчиками Linux, признающими особую роль FSF в мире Open Source. Но это еще не говорит о том, что любые обновления GPLv2 должны полностью соответствовать такой договоренности.

## Пункт 5. GPLv3

В последнем черновом варианте GPLv3 появился целый ряд проблем, которых не было в предыдущей версии лицензии:

### Условия DRM

Несмотря на то, что использование DRM медийными компаниями настораживает, это не повод лишать других их свобод. Попытки FSF варьировать этими ограничениями приводят к устранению возможности безобидного и выгодного использования технологий шифрования и DRM. Такие условия опасны, и их трудно сделать правильными, поэтому их попросту не должно быть в обновлении к GPLv2.

Кроме того, определение ограничений по DRM — слишком политическое по своей природе. Попытки представить подобные условия в GPLv3 и заново лицензировать весь код FSF равносильны подведению трудов всех, кто принимал участие в проектах, к политическим убеждениям FSF.

### Дополнительные ограничения

Раздел текущего шаблона лицензии с дополнительными ограничениями приведет лишь к очередным проблемам с лицензиями у поставщиков Linux-дистрибутивов, которым придется разбираться с совместимостью лицензий всех компонентов системы. Кроме того, это может привести

к разделению лицензий на отличающиеся по типу выбранных ограничений.

### Патентные положения

Текущая версия лицензии фактически говорит о том, что она подвергнет опасности все портфолио патентов у любой компании, которая опубликует на своем сайте программу, лицензированную под GPLv3. Это недопустимо для случая с Linux, где компании существенно помогают развитию проекта.

К тому же у некоторых нынешних Linux-поставщиков есть солидный набор патентов, а такие ограничения явятся дополнительным барьером для продвижения их версий GNU/Linux, что критично для всей экосистемы.

## Выводы

3 основные причины, обозначенные в пункте 5, уже являются достаточными для того, чтобы не принимать нынешнюю версию GPLv3 для Linux-ядра. Нет и никаких надежд на то, что будущая редакция лицензии GPLv3 станет приемлемой для Linux.

Более того, поскольку FSF предлагает всем своим проектам переходить на GPLv3 и оказывает давление на каждый из них, этот процесс может привести к многочисленным разветвлениям среди проектов (к «балканизации»), а в конечном итоге — нанести непоправимый вред всей экосистеме Open Source. Учитывая же отсутствие достаточной ценности в текущих редакциях лицензии GPLv3, это явно неоправданный ход.

Разработчики Linux-ядра настоятельно рекомендуют FSF пересмотреть возможные последствия их действий и избежать текущего процесса GPLv3, пока не слишком поздно.

## Реакция FSF

Организация свободного ПО решила не оставаться в стороне от критики, поступившей в адрес ее проекта, и 25 сентября опубликовала ответ, призванный разъяснить «неточные сведения» о втором черновом варианте GNU GPL 3 (в частности и в первую очередь он относится к разработчикам Linux-ядра).

FSF постаралась в документе (<http://www.fsf.org/news/gplv3-clarification>), состоящем из трех пунктов, уточнить некоторые моменты о новой лицензии, которые, по ее мнению, были неправильно поняты:

- ☑ У FSF нет прав на то, чтобы заставить все проекты перейти с GPLv2 на GPLv3. Это дело каждого, и вмешиваться в решения, принятые разработчиками, организация не собирается.
- ☑ Во избежание нарушения свободы 0



(право запускать программу, как это угодно пользователю), в черновом варианте GPLv3 нет никаких ограничений на использование ПО, как некоторые об этом сообщали.

- ❑ Если GPLv2 полагается на не выраженные прямо патентные лицензии, зависящие от законодательства США,

то GPLv3 ясно указывает на патентные лицензии, применимые ко всему миру. Это вовсе не значит, что компания может «потерять все свое портфолио патентов», а лишь говорит о том, что если у кого-то есть патенты, затрагивающие XYZ, и он распространяет GPL-ПО для выполнения XYZ, то он не сможет

подать иск на последующих пользователей, распространителей программы и желающих улучшить ее за выполнение XYZ с помощью их собственных версий этой программы.

Дмитрий Шурупов,  
(osa@samag.ru)

## Проект XGL: что нового?

С момента анонса Xgl прошло уже 9 месяцев – много всего за это время изменилось. В статье приведен краткий обзор последних новшеств.

На форуме [compiz.net](http://compiz.net) (а теперь уже на [forum.beryl-project.org](http://forum.beryl-project.org)) активно ведется разработка другой версии Compiz – Beryl, где появляются самые прогрессивные идеи и их реализации.

Посмотрим, что же несет нам Beryl и выясним, откуда пришла идея того или иного плагина.

Начнем с недавнего улучшения, которого, наверное, ждали очень многие, – избавление от gconf. Теперь нет необходимости ставить gconf, чтобы получить работающий Beryl. Плагин, заменяющий gconf, называется csm, а также есть программа, осуществляющая конфигурацию Beryl – *beryl-manager* (см. **рис. 1**).

Плагинов же вообще появилось большое множество, выделю среди них следующие:

- ❑ **Animation** – позволяет применить различные эффекты для действий – появление/исчезание окон, сворачивание/разворачивание, получение фокуса окном (см. **рис. 2**). Данный плагин является заменой *minimize*, идеи некоторых эффектов заимствованы из OSX.
- ❑ **Benchmark** – не самый полезный плагин, показывающий текущее количество прорисовываемых кадров в секунду.
- ❑ **Blur/Reflection** – первый плагин позволяет сделать фон под прозрачными декорациями окна смазанным (см. **рис. 4**), а второй – наложить текстуру между декорацией и низлежащим фоном (см. **рис. 5**). В Beryl эти плагины были заменены одним – *blurfx*, который также предоставляет и функцию *motion blur* (эффект похож на активацию азота в игре NFS) (см. **рис. 3**).
- ❑ **Crash handler** – весьма полезный плагин; позволяет Compiz автоматически перезапускаться при ошибке сегментации данных или запускать другой оконный менеджер.

- ❑ **Negative** – весьма забавный плагин, позволяющий инвертировать цвета в окне, или на всем экране.
- ❑ **Put** – позволяет задать, в какой области экрана будут появляться вновь созданные окна, а также перемещать окно по текущему экрану и на другие экраны, не вращая кубик (кстати, появилась и возможность отдалять кубик перед поворотом или перемещаться по экранам, как по слайдам – влево-вправо).
- ❑ **Showdesktop** – очень полезный плагин, позволяющий расположить окна по краям рабочего стола, освободив таким образом его большую часть (идея пришла напрямую из MacOS X).
- ❑ **Screenshot** – плагин предназначенный для простого создания снимков как всего экрана, так и отдельной его части.
- ❑ **State** – достаточно удобный плагин, позволяющий настроить уровень прозрачности, яркости, контрастности, места появления на рабочем столе, экрана появления для различных типов окон (например, можно сделать так, чтобы все меню были полупрозрачными).

Кроме того, участниками форума был разработан новый декоратор, названный *emerald*. Он поддерживает плагины, каждый из которых по-своему прорисовывает декорацию окна, и темы. Из заметных возможностей *emerald* – настраиваемые кнопки (из файлов PNG), их расположение, подсветка вокруг них при наведении (см. **рис. 6**). Также можно установить размеры и цвета границ, мигание кнопок при наведении и много другое. Все это делается в программе *emerald-theme-manager*. На сегодняшний момент существует несколько десятков тем.

Если посмотреть на изменения за прошедшие полгода с момента предыдущего обзора Compiz в «Open Source», можно сказать, что начался процесс реализации желаемых идей, а недостаток в функционале становится все меньше и меньше.

Сообщество форума [compiz.net](http://compiz.net) смогло создать свою версию Compiz – Beryl, намного более функциональную, чем оригинальная. Ответвление Beryl от Compiz было вполне ожидаемым событием, т.к. *compiz-quinntorm* достаточно сильно отличался от своего родителя.

Остается ждать улучшения стабильности и работы над ошибками.

**P.S.** Во время написания данной статьи вышел Beryl 0.1.0. Релиз работает на Xgl/AIGLX/XOrg 7.1 (с драйверами

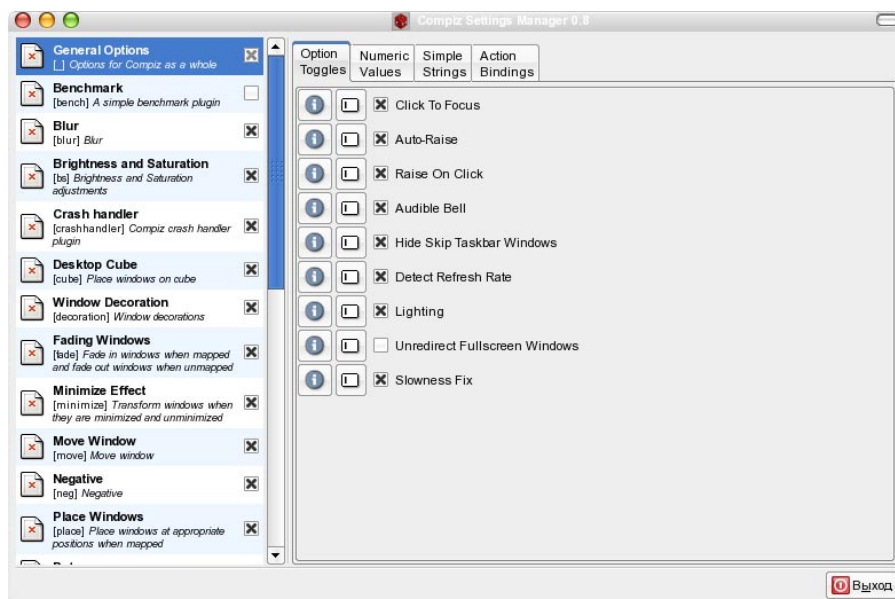


Рисунок 1

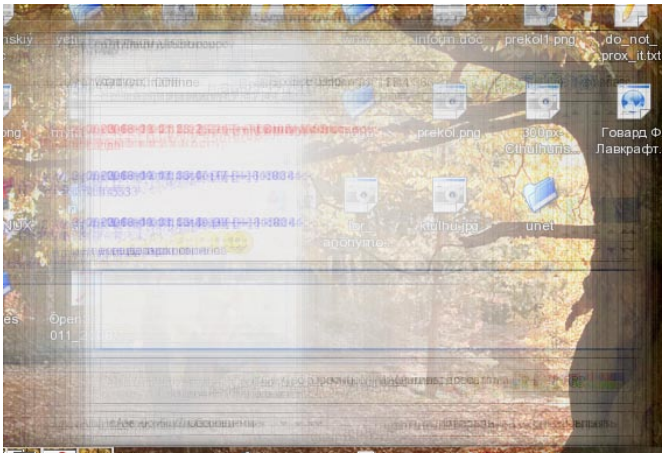
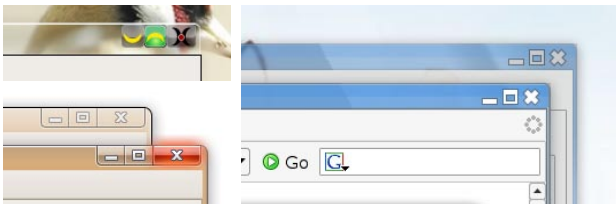


Рисунок 2



Рисунок 3



Рисунки 4, 5, 6

что драйверы NVIDIA используют AIGLX, но это не так. Если взглянуть на архитектуру AIGLX (см. «Ускорение GLX: AIGLX и Xgl» в «Open Source» 007), то становится ясно, что для ра-

боты требуется наличие открытого драйвера DRI – в этих драйверах собственная реализация ускоренного непрямого рендеринга (за него отвечает расширение nv-glx). Более подробно узнать о том, что добавить в xorg.conf для работы с Beryl, можно узнать на <http://lists.freedesktop.org/archives/compiz/2006-September/000462.html>.

NVIDIA 9625). Последнее следует разъяснить. Многие неправильно полагают,

работы требуется наличие открытого драйвера DRI, а NVIDIA не использует архи-

Владимир Кемпик  
([bobahu4@mail.ru](mailto:bobahu4@mail.ru))

## Обзор музыкального демона mpd

Статья посвящена необычному аудиоплееру – mpd (Music Player Daemon). Этот демон позволяет воспроизводить музыку исходя только из плейлиста.

### Настройка и запуск mpd

Архивы с mpd можно найти на сайте проекта – <http://www.musicpd.org>. После установки mpd обратимся к настройкам сервера, которые содержатся в единственном файле – /etc/mpd.conf. Вот содержимое моего «рабочего» конфигурационного файла:

```
$ cat /etc/mpd.conf | egrep -v '^(#|$)'

music_directory "/mnt/mine/media/music"
playlist_directory "/var/lib/mpd/playlists"
db_file "/var/lib/mpd/database"
log_file "/var/log/mpd/mpd.log"
error_file "/var/log/mpd/mpd.error.log"
pid_file "/var/run/mpd/mpd.pid"
filesystem_charset "UTF-8"
state_file "/var/lib/mpd/state"
user "mpd"
bind_to_address "localhost"
```

Как видно из названия, music\_directory – путь к вашей музыкальной коллекции. Все остальные параметры я оставил по умолчанию, но конфиг имеет хорошие комментарии, так что вам будет несложно разобраться.

После изменения настроек необходимо произвести обновление базы и можно запускать mpd.

Базу данных демон ведет по музыкальной коллекции и плейлистам. Ее обновление происходит только по команде, поэтому если вы добавили какую-либо группу в свою коллекцию (или готовите mpd к первому запуску в системе), необходимо запустить mpd с ключом --create-db:

```
# /usr/bin/mpd --create-db
```

Если процесс mpd уже был запущен, то необходимо сделать его рестарт:

```
# /etc/init.d/mpd restart
```

В ином случае – запустить:

```
# /etc/init.d/mpd start
```

Управление демоном осуществляется по сети – mpd слушает определенный порт в ожидании команд для выполнения. Чтобы совершить какие-либо действия (например, составить плейлист или поставить на проигрывание файл), необхо-

димо воспользоваться клиентской программой.

### Оболочки к mpd и работа с ними

У mpd существует достаточное число оболочек (front-end): есть и под GTK+, и под Qt (3 и 4 версий), и для консоли, и с веб-интерфейсом. Они подключаются к mpd к оп-

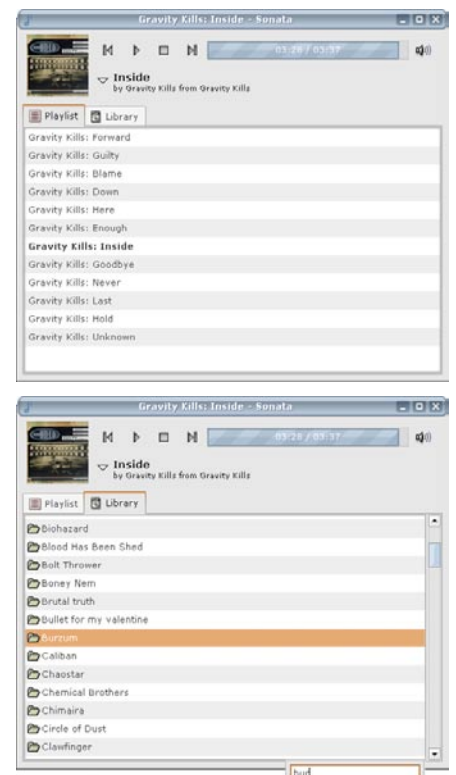


Рисунок 1. Sonata

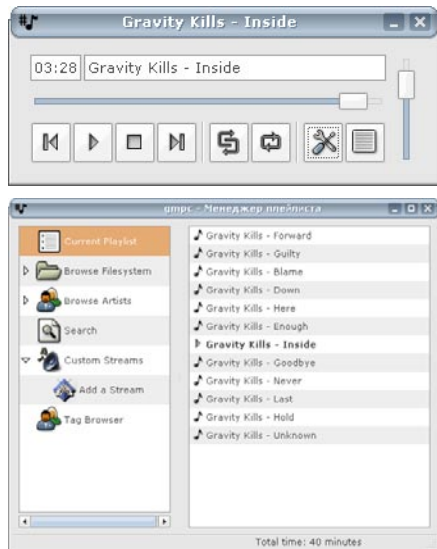


Рисунок 2. Gmpc

ределенному порту, считывают информацию о состоянии mpd и плейлист. Составив список песен и поставив их на проигрывание, вы можете закрыть оболочку, выйти из X-ов, а музыка будет продолжать проигрываться.

Сначала эта особенность мне казалась не особо интересной, но теперь я понимаю все удобство данной схемы. К тому же, если у вас дома технопарк, вы можете использовать front-end с любого компьютера, подключаясь по сети. Рассмотрим 6 оболочек.

## Sonata

Основанный на GTK+ клиент для mpd.

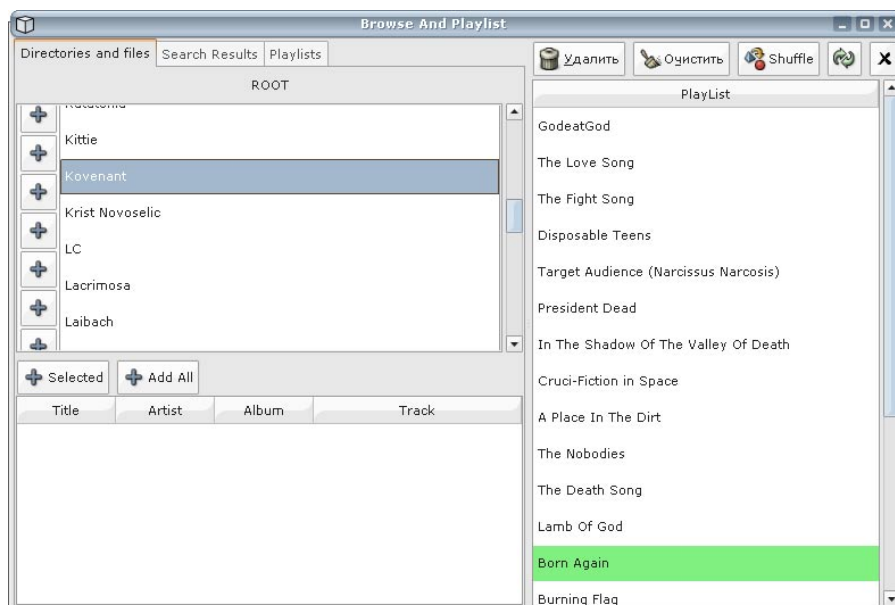


Рисунок 4. GtK2mp

Сначала он мне показался немного медлительным, так как стартует дольше остальных, но потом я привык и довольно долго пользовался только им.

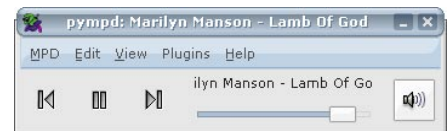
Список особенностей:

- ☒ Плейлист.
- ☒ Обзор базы данных.
- ☒ Поиск в базе по первым буквам названия.
- ☒ Режим компактного вида.
- ☒ Просмотр обложек.

Так как оболочка большую часть времени не включена, неудобство отсутствия иконки в системном трее у Sonata не ощущается.



Рисунок 3. Pympd



Сайт – <http://developer.berlios.de/projects/sonata>.

## Gmpc

Стандартный GNOME-клиент для mpd.

Список особенностей:

- ☒ Плейлист.
  - ☒ Обзор базы данных.
  - ☒ Поиск в базе по первым буквам названия.
  - ☒ Древовидное представление базы.
  - ☒ Сортировка базы по множеству параметров (очень удобная возможность при сортировке по стилям).
  - ☒ Поддержка выборочных потоков.
  - ☒ Просмотр обложек (хотя я его и не нашёл).
  - ☒ Поддержка трее, показывает уведомления.
- Возможностей больше и работает побыстрее.

Сайт – [http://cms.qballcow.nl/index.php?page=Gnome\\_Music\\_Player\\_Client](http://cms.qballcow.nl/index.php?page=Gnome_Music_Player_Client).

## GtK2mp

Оболочка к mpd на базе GTK2. Так же, как и у Gmpc, у этого клиента плейлист отделен от главного окна.

Список особенностей:

- ☒ Плейлист.



- ✓ Обзор базы данных.
- ✓ Поиск в базе по группе/альбому/названию песни и названию файла.
- ✓ Редактор плейлистов.

Эта оболочка не понравилась мне своей работой с плейлистами. Так как нет поддержки drag'n'drop и соответствующих кнопок управления, я так и не понял, как менять последовательность треков.

К тому же при просмотре коллекции кнопки добавления каталогов кое-где съезжают, что выглядит очень некрасиво.

Сайт – <http://www.musicpd.org>.

## Рympd

Клиент для mpd на базе PyGTK+, выполненный в стиле Rhythmbox.

Список особенностей:

- ✓ Плейлист.
- ✓ Обзор базы данных.
- ✓ Сортировка по тегам.
- ✓ Редактор горячих клавиш (не глобальные).
- ✓ Несколько плагинов в комплекте (в том числе и поддержка системного трея).
- ✓ Режим компактного вида.
- ✓ Редактор плейлистов.
- ✓ Фильтр плейлиста.

Эта оболочка понравилась мне больше всех. Правда, проблема с изменением последовательности треков в плейлисте аналогична той, что наблюдается у Gtk2mpd, да и скорость работы не выше, чем у Sonata.

Сайт – <http://sourceforge.net/projects/rympd>.

## Qmpdclient

Клиент для mpd на базе Qt 4.1.

Список особенностей:

- ✓ Плейлист.
- ✓ Обзор базы данных.
- ✓ Сортировка по тегам.
- ✓ Древоподобное представление базы.
- ✓ Поддержка интернет-радио.
- ✓ Поиск в базе по первым буквам названия.
- ✓ Поддержка трея, показывает уведомления.
- ✓ Редактор плейлистов.
- ✓ Фильтр плейлиста.

Скорость работы очень обрадовала, учитывая, что это один из самых функциональных клиентов: стартует мгновенно и откликается без задержек. Были некоторые проблемы со сборкой, но скорее всего, они вызваны тем, что я использую qt-4.2.0\_rc1.

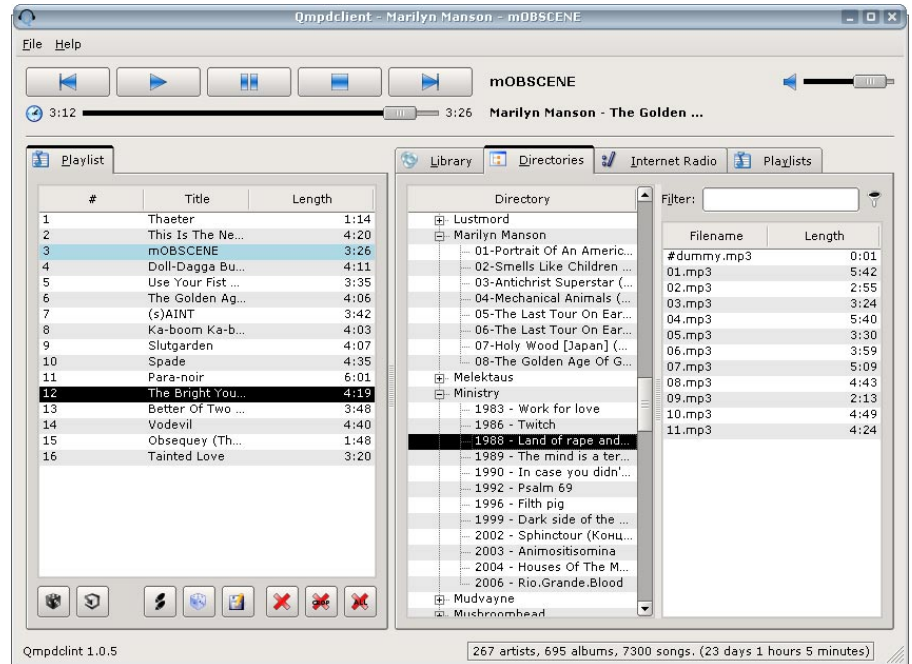


Рисунок 5. Qmpdclient

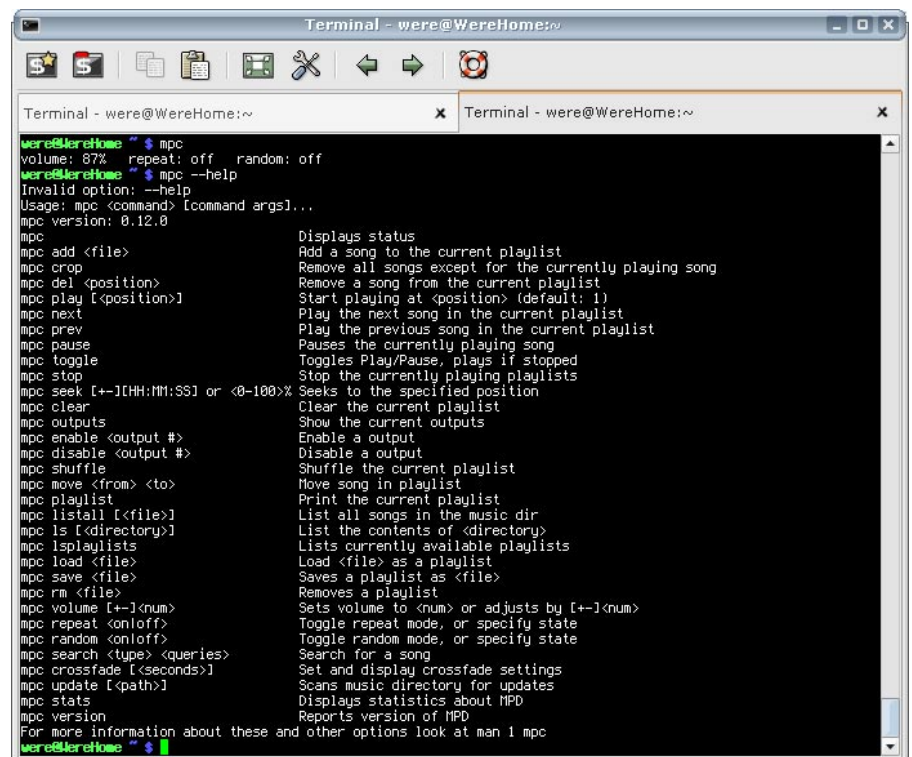


Рисунок 6. Mpc

Сайт – <http://havtknut.tihlde.org/software/qmpdclient>.

## Mpc

Управление этим плеером осуществляется только через командную строку. Очень полезная находка для любителей все вешать на горячие клавиши.

Сайт – <http://www.musicpd.org>.

## Итоги

Любителям многофункциональных программ с кучей настроек советую попробовать Gmpc и Qmpdclient. Также Rympd в этом плане очень неплох. Если же вы

любитель чего «попроще», обратите внимание на Sonata.

В данном обзоре я рассмотрел не все клиенты. С их списком можете ознакомиться по адресу: <http://mpd.wikia.com/wiki/Clients>.

Для обладателей малых объемов оперативной памяти mpd может оказаться очень полезной находкой, так как поставив музыку на проигрывание, вы можете выйти из самой оболочки, а демон потребляет очень мало памяти.

Алексей Попов  
([gnu.troll@gmail.com](mailto:gnu.troll@gmail.com))

## Виртуальные серверы в GNU/Linux на базе Linux-VServer

Если становится сложно справляться с администрированием множества служб на одном сервере или вы не хотите давать привилегии суперпользователей другим администраторам, попробуйте распределить службы по виртуальным серверам. В наше время неудержимого прогресса информационных технологий и развития глобальных сетей многим системным администраторам приходится сталкиваться с задачами, связанными с созданием ресурсов для широкого пользования. С увеличением объема таких ресурсов, находящихся в руках одного администратора, растет сложность администрирования, и, как правило, страдает безопасность. Встают вопросы частичного делегирования административных полномочий. Техники VPS (Virtual Private Server) и VDS (Virtual Dedicated Server) позволяют решать такие задачи – задачи «эффективного разделения целого на меньшее». В отличие от других небезызвестных технологий, таких как VMWare и Xen, VPS предоставляет минимальную виртуализацию. Подробнее различные виды виртуализации описаны здесь: <http://en.wikipedia.org/wiki/Virtualization>.

В этой статье рассматривается технология Linux-VServer (<http://linux-vserver.org>). Описываются основные возможности, а также вы найдете инструкции по установке программного обеспечения и настройке виртуальных серверов.

### Основные возможности

Технология Linux-VServer позволяет создавать несколько виртуальных серверов (VPS), работающих независимо под управлением одного ядра операционной системы. Каждый виртуальный сервер связан с одним контекстом, в котором выполняются его процессы. Контекст – это окружение, объединяющее группу процессов в системе, и отделяющее их от процессов, не входящих в этот контекст. Имена пользователей и групп, а также их идентификаторы принадлежат контексту. Процессы главного (корневого) сервера входят в контекст с нулевым номером и обладают большими возможностями, они имеют доступ к процессам и данным всех виртуальных серверов. В файловой системе главного сервера корни файловых систем виртуальных серверов расположены в некоторой директории.

VPS внутри выглядит как полноценная Linux-система. Вы можете предоставить права root на него и совершенно не бояться, что как-то могут быть повреждены другие виртуальные серверы или главный сервер. Все сервисы, такие как электронная почта, базы данных, Web, SSH, могут быть запущены без модификаций (или с некоторыми минимальными модификациями) на виртуальном сервере (список программ, с которыми возникают проблемы, вы можете посмотреть здесь: <http://wiki.linux-vserver.org/ProblematicPrograms>). Каждый VPS может обладать как одним IP-адресом, так и несколькими. Для виртуального сервера может быть настроено множество параметров и ограничений. Вот наиболее часто используемые из них:

- ☑ Место, занимаемое на диске, и количество индексных дескрипторов (файлов).
- ☑ Объем физической и виртуальной памяти.
- ☑ Доля использования мощности процессора и список процессов, на которых может выполняться VPS.

- ☑ Максимальное количество процессов, принадлежащих контексту.

Для каждого VPS могут отслеживаться такие показатели, как:

- ☑ Использованное процессорное время.
- ☑ Количество совершенных системных вызовов fork (с его помощью создаются новые процессы).
- ☑ Количество полученных и переданных сетевых пакетов.

Программное обеспечение

- ☑ Операционная система GNU/Linux; Linux-ядро 2.6.17 (<http://kernel.org>).
- ☑ Linux-VServer Kernel Patches 2.0.21 (<http://wiki.linux-vserver.org/Downloads>).
- ☑ Linux-VServer Userspace Utilities 0.30.210 (<http://wiki.linux-vserver.org/Downloads>).

Также потребуется образ, который будет использоваться на VPS. Вот некоторый список архивов, содержащих готовые образы:

- ☑ <http://mirrors.sandino.net/vserver/images>;
- ☑ <http://debian.marlow.dk/vserver/guest>;
- ☑ <http://www.fumanchu.com/vserver>;
- ☑ <http://lylix.net/vps+templates/func.select/id.1>;
- ☑ <http://distfiles.gentoo.org/experimental/x86/vserver>;
- ☑ <http://distfiles.gentoo.org/experimental/amd64/vserver>.

Отличается такой образ от полноценной Linux-системы обычно измененными скриптами инициализации, а в особенности – скриптами, направленными на проверку целостности корневой файловой системы и т. п.

Разумеется, никто не отменяет возможности самостоятельного создания образа на основе используемого вами дистрибутива.

### Установка поддержки Linux-VServer в ядре

Скачаем ядро, разархивируем его и добавим патч:

```
# wget http://www.kernel.org/pub/linux/kernel/v2.6/ linux-2.6.17.13.tar.bz2
# wget http://ftp.linux-vserver.org/pub/kernel/vs2.0/ patch-2.6.17.13-vs2.0.2.1.diff.bz2
# tar -xjf linux-2.6.17.13.tar.bz2
# bunzip2 patch-2.6.17.13-vs2.0.2.1.diff.bz2
# cd linux-2.6.17.13
# patch -p1 < ../patch-2.6.17.13-vs2.0.2.1.diff
# make menuconfig
```

После того, как вы примените патч к ядру, в меню конфигурации появится новая секция «Linux-VServer», ей мы и уделим внимание.

Опции, отвечающие за совместимость со старым управляющим программным обеспечением и не представляющие в нашем случае никакого интереса. Их надо отключить.

```
Linux-VServer -> Show a Legacy Version Number (VSERVER_LEGACY_VERSION=n)
Linux-VServer -> Enable Legacy Kernel API (VSERVER_LEGACY=n)
```

Следующая опция отвечает за устаревшую поддержку сети в виртуальных серверах, эта поддержка, как написано в документации, нужна:

```
Linux-VServer -> Disable Legacy Networking Kernel API (CONFIG_VSERVER_NGNET=n)
```

Опция, отвечающая за то, чтобы виртуальные серверы видели только свои процессы (в отличие от главного сервера, который может просматривать процессы всех VPS (с помощью специальных утилит):



```
Linux-VServer -> Enable Proc Security (VSERVER_PROC_SECURE=y)
```

Опция, отвечающая за возможность жесткого управления процессами. Если VPS превысил свой лимит процессорной мощности и для него установлено жесткое управление, выполнение процессов контекста будет заморожено на некоторый промежуток времени (подробнее смотреть «Ограничение ресурсов»):

```
Linux-VServer -> Enable Hard CPU Limits (VSERVER_HARDCPU=y)
```

Следующая опция отвечает за ограничение времени проста контекста, она позволяет добиться большей интерактивности (меньшего времени отклика), но и сильно увеличивает перегрузку, связанную с планированием процессов (отправкой их на выполнение). Оставляю выбор на ваше усмотрение:

```
Linux-VServer -> Limit The IDLE task (VSERVER_HARDCPU_IDLE)
```

Опция, отвечающая за разделение битов в идентификаторах пользователя и группы на биты, характеризующие номер контекста, и биты, являющиеся идентификатором пользователя внутри VPS. Изменяя эти числа, можно достичь разного максимального числа возможных групп и пользователей в системе. Используется при сохранении принадлежности файла к конкретному VPS в файловой системе. Рекомендую оставить значение по умолчанию:

```
Linux-VServer -> Persistent Inode Context Tagging -> UID24/GID24 (INOXID_UGID24=y)
```

Опция, отвечающая за поддержку встроенного в ядро NFS-демона:

```
Linux-VServer -> Tag NFSD User Auth and Files (XID_TAG_NFSD=n)
```

После настройки этих опций скомпилируем и установим ядро, поправим соответствующим образом загрузчик, а потом не забудем перезагрузиться:

```
# make bzImage modules modules_install install
```

## Устанавливаем программное обеспечение для управления VPS

Для работы утилит нам понадобится программа vconfig (<http://www.candelatech.com/~greear/vlan.html>). Скачаем и разархивируем утилиты:

```
# wget http://ftp.linux-vserver.org/pub/utlils/utlil-vserver/ \
  util-vserver-0.30.210.tar.bz2
# tar -xjf util-vserver-0.30.210.tar.bz2
# cd util-vserver-0.30.210
```

Конфигурацию производим следующим образом:

```
# ./configure --prefix=/usr/local/vserver \
  --with-vrootdir=/var/vservers
```

Опция «vrootdir» указывает папку, где будут располагаться корни виртуальных файловых систем. Компилируем и устанавливаем:

```
# make
# make install
```

Дополним переменные окружения новыми путями (это лучше прописать, например, в `bashrc`):

```
# PATH=$PATH:/usr/local/vserver/sbin
# MANPATH=$MANPATH:/usr/local/vserver/man
```

Настроим файловую систему, в которой будут располагаться виртуальные серверы (в ней будет находиться `vrootdir`, в нашем случае – `/var/vservers`).

При монтировании нам нужно обязательно указать опцию «tagxid»: она включает установку принадлежности файлов конкретному VPS (см. «Ядро: Linux-VServer -> Persistent Inode Context Tagging»).

Также, если вы используете `reiserfs`, вам понадобится опция «attr», включающая поддержку атрибутов файла (`chattr`, `lsattr`) (проверьте, включена ли в ядре опция `REISERFS_FS_XATTR` при использовании `reiserfs`).

Монтирование производится следующим образом:

```
# mount /dev/sda2 /var/ -o tagxid,attr
```

Или, в случае `fstab`, это будет строка вида:

```
/dev/sda2    /var    reiserfs    tagxid,attr    1 1
```

В этом примере монтирование производится в директорию `/var`, это необязательно.

После установки поддержки в ядре, установки утилит, настройки файловой системы произведем перезагрузку.

Создаем директорию, в которой будут храниться наши VPS, и устанавливаем для нее флаг «Chroot Barrier» («барьер смененного корня»). Отвечает этот флаг за то, что никакой процесс из VPS (контекста) не сможет выбраться за этот барьер.

```
# mkdir /var/vservers
# /usr/local/vserver/sbin/setattr --barrier /var/vservers/
```

Для продолжения понадобится выполнить команду, делающую видимыми для самого сервера некоторые части `/proc` (см. «Ядро: Linux-VServer -> Enable Proc Security»). Полезно добавить эту команду к скриптам инициализации системы.

```
# /usr/local/vserver/etc/init.d/vprocunhide start
```

Для того чтобы перезагрузка и выключение виртуальных серверов обрабатывались корректно, потребуется сделать следующее:

```
# echo 'kernel.vshelper=/usr/local/vserver/lib/ \
  util-vserver/vshelper' >> /etc/sysctl.conf
# sysctl -p
```

Следующая команда должна вывести информацию о версии Linux-VServer API в ядре и версии утилит. Если она делает это, то установка прошла успешно.

```
# vserver-info
```

## Создание VPS

Для начала нужен образ VPS. Мы выберем для себя образ на основе Gentoo. Скачаем и распакуем его:

```
# wget http://distfiles.gentoo.org/experimental/x86/vserver/ \
  stage3-i686-20060317.tar.bz2
# mkdir /usr/src/stage3-vserver
# tar -xjf stage3-i686-20060317.tar.bz2 -C /usr/src/stage3-vserver
# mkdir /usr/src/stage3-vserver/usr/portage
```

Если вы используете не Gentoo в качестве дистрибутива для главного сервера, вам понадобится скачать и установить дерево портежей (это дерево можно можно монтировать к нескольким VPS для совместного использования):

```
# wget http://distfiles.gentoo.org/snapshots/portage-latest.tar.bz2
# tar -xjf portage-latest.tar.bz2 -C /usr
```

Создадим базовую конфигурацию для нашего нового VPS:

```
# vserver testvps build -m skeleton --context 5 --initstyle plain
```

Рассмотрим параметры команды:

- ✓ **testvps** – имя виртуального сервера.
- ✓ **-m skeleton** – метод, означающий, что надо создать только конфигурационные файлы.
- ✓ **--context 5** – идентификатор контекста.
- ✓ **--initstyle plain** – стиль загрузки (<http://oldwiki.linux-vserver.org/InitStyles>).

После выполнения команды для нового сервера создана конфигурация в `/usr/local/vserver/etc/vservers/testvps`, зайдём в эту директорию.

Скопируем файловую систему нашего будущего VPS:

```
# cp /usr/src/stage3-vserver/* /var/vservers/testvps/ -pPR
```

Дополним `fstab` следующим образом:

```
# cat >> /usr/local/vserver/etc/vservers/testvps/fstab
/usr/portage /usr/portage none bind,ro 0 0
/usr/portage/distfiles /usr/portage/distfiles none bind,rw 0 0
```

Первая строка монтирует «дерево портежей» в режиме «только чтение», вторая – директорию, содержащую архивы исходных файлов, используемые пакетным менеджером `emerge`.

Запустим наш новый виртуальный сервер:

```
# vserver testvps start
```

Посмотрим информацию о запущенных серверах:

```
# vserver-stat
```

CTX	PROC	VSZ	RSS	userTIME	sysTIME	UPTIME	NAME
0	106	1.6G	744.4M	36m41s10	6m39s70	17h45m58	root server
5	1	1.4M	492K	0m00s21	0m00s13	0m00s75	testvps

Обратите внимание, что главный сервер представляется как VPS с номером 0 и именем «root server».

Для входа в консоль VPS можно использовать следующую команду:

```
# vserver testvps enter
```

## Настройка сети в VPS

Прежде всего перейдем в каталог с конфигурацией интерфейсов виртуального сервера и создадим директорию для первого интерфейса:

```
# cd /usr/local/vserver/etc/vservers/testvps/interfaces
# mkdir 0
# cd 0/
```

Установим имя устройства, к которому будет присоединен IP-адрес нашего VPS:

```
# echo 'eth0' > dev
```

Укажем псевдоним устройства, создаваемого при инициализации VPS:

```
# echo 'testvps' > name
```

Настроим IP-адрес и маску сети:

```
# echo '192.168.0.2' > ip
# echo '24' > prefix
```

Перезагрузим VPS:

```
# vserver testvps restart
```

Для того чтобы посмотреть получившийся результат, воспользуемся командой `ip` (входит в `iproute2`: <http://linux-net.osdl.org/index.php/Iproute2>).

```
# ip addr | grep 192.168.0.1
```

На экране мы увидим:

```
inet 192.168.0.2/24 brd 192.168.0.255 scope global eth0: testvps
```

Обратите внимание, что полное имя устройства вида <имя>:<псевдоним> не может быть в длину больше 15 символов. В VPS можно увидеть результат:

```
# vserver testvps enter
# ifconfig
```

Выход из консоли VPS осуществляется стандартно: комбинацией клавиш `<Ctrl+D>` или командой `logout`.

Если по аналогии создать второй виртуальный сервер и присвоить ему IP-адрес из той же сети (например, 192.168.0.3) серверы смогут осуществлять коммуникации друг с другом.

## Архитектурные рекомендации

Наиболее востребованной технология VPS оказалась для создания хостинговых решений. Большую актуальность она имеет и в образовательной среде, так как предоставляет возможность без последствий давать привилегии суперпользователя (`root`). А как без них обучить администрированию?

Рассмотрим четыре способа организации доступа к виртуальным серверам. Некоторые из них могут быть использованы для организации публичного (для неограниченной аудитории) доступа, другие – для групп пользователей конечного размера. Выбирая наиболее эффективные варианты доступа в конкретной ситуации и правильно комбинируя их, можно добиться больших гибкости, удобства использования и надежности.

### Публичный IP на каждый VPS

Это самый простой вариант для реализации и в то же время самый расточительный для адресного пространства IP. заключается он в том, что на каждый VPS выделяется один (или более) публичный IP. Таким образом, любой сервис, запущенный на VPS и слушающий на каком-то порту, будет доступен из глобальной сети.

### Приватный IP + Port Forwarding (DNAT)

Этот вариант очень удобен, когда вы хотите предоставить доступ к одному или нескольким сервисам, запущенным на VPS, например, для администрирования.

VPS-серверам выделяются IP-адреса из приватного диапазона некоторой виртуальной сети. Для нужных сервисов на VPS-серверы пробрасываются порты с публичных IP-адресов, привязанных к самому серверу. Таким образом, обращаясь на один и тот же IP-адрес, например, по портам 21 (FTP) и 80 (HTTP), мы можем в действительности обращаться к разным VPS.

Основным недостатком такой реализации является то, что она совершенно не пригодна для HTTP-хостинга. Скорее все-

го, вас не поймут, если вы предложите одним своим клиентам обращаться к своему сайту по порту 1080, а другим – по 8080, что простительно в случае с административным доступом.

## Приватный IP + VPN

Так же, как и в предыдущем варианте, VPS назначаются приватные IP-адреса. В получившуюся приватную сеть организовывается VPN-доступ с использованием, например, PPTP или OpenVPN, или IPSec.

Этот вариант может быть использован не для публичного доступа, а для административного или для доступа некоторой небольшой группы людей. Многие VPN-реализации предоставляют шифрование: таким образом, работая через VPN с сервисами, запущенными на VPS, можно гарантировать некоторый, достаточно высокий, уровень сохранности данных. В образовательной среде этот вариант имеет большие преимущества, так как позволяет малыми затратами адресного пространства сделать многое.

## Приватный IP + HTTP-прокси

На главном сервере или на одном из VPS с публичным IP-адресом мы запускаем реверсный HTTP-прокси-сервер, перенаправляющий в зависимости от доменного имени запросы внутри виртуальной сети на нужные VPS. Таким образом, через 80 порт на одном публичном IP мы можем предоставлять HTTP-доступ к сайтам, расположенным на разных VPS. В качестве реверсного прокси-сервера прекрасно подходит продукт российского производства nginx (<http://sysoev.ru>).

Этот вариант, к большому сожалению (в силу внутреннего устройства протокола), пригоден только для HTTP.

## Настройка ограничений дискового пространства

Для настройки перейдем в директорию с конфигурацией VPS и создадим там каталог:

```
# cd /usr/local/vserver/etc/vservers/testvps/
# mkdir dlimits
# mkdir dlimits/0
# cd dlimits/0
```

Укажем директорию, для которой мы устанавливаем ограничения:

```
# echo '/var/vservers/testvps' > directory
```

Ограничим количество индексных дескрипторов файлов:

```
# echo '100000' > inodes_total
```

Ограничим пятью гигабайтами место, доступное VPS:

```
# echo '5242880' > space_total
```

Установим процент от общего размера, резервируемый для пользователя root.

```
# echo '5' > reserved
```

После этого перезагрузим VPS, войдем в него и посмотрим, что получилось:

```
# df -h
```

```
/dev/hdv1 5,0G 0 4,8G 0% /
```

Так как для файлов образа не установлен идентификатор контекста (XID), место на диске, занимаемое ими, не учитывается. Если вы хотите, чтобы и оно учитывалось, можно сделать следующее:

```
# vserver testvps stop
# chxid -c 5 -R /var/vservers/testvps
# vserver testvps start
```

Входим и проверяем:

```
# df -h
```

```
/dev/hdv1 5,0G 503M 4,3G 11% /
```

## Настройка ограничений ресурсов

Перейдем в директорию с конфигурацией виртуального сервера:

```
# cd /usr/local/vserver/etc/vservers/testvps/
# mkdir rlimits
# cd rlimits
```

Существуют следующие ресурсы: cpu, fsize, data, stack, core, rss, nproc, nofile, memlock, as, locks. Более подробно о типах ресурсов можно прочитать в `man setrlimit`.

В директории создаются файлы с именами <имя ресурса>.min, <имя ресурса>.soft, <имя ресурса>.hard, в которых и определяются конкретные ограничения для ресурсов.

В файле `/usr/local/vserver/etc/vservers/testvps/flags` вы можете установить флаги, рассмотрим некоторые часто используемые:

- ☒ **virt\_uptime** – показывать внутри VPS время работы виртуального сервера.
- ☒ **virt\_load** – показывать индикатор загрузки VPS, а не главного сервера.
- ☒ **sched\_hard** – указывает планировщику замораживать процесс, если он превысил лимит мощности процессора.
- ☒ **sched\_prio** – указывает планировщику понижать приоритет процесса, если он превысил лимит процессора.
- ☒ **fork\_rss** – запрещать системный вызов `fork`, если превышен `rlimit rss`.

В файле `/usr/local/vserver/etc/vservers/schedule` вы можете установить параметры планировщика. Например, для того чтобы VPS выделялось 7/32 (21.9%) каждого процесса, а в случае превышения он приостанавливался на  $(200/7)*32=928$  так называемых мгновений (jiffies), нужно создать файл следующего содержания:

```
7
32
500
200
1000
string
```

928 jiffies – это порядка одной секунды, если частота таймера равна 1000 Гц и примерно десять секунд, если частота равна 100 Гц (частота таймера в конфигурации ядра: «Processor type and features -> Timer frequency»). Управление планировщиком находится в стадии разработки и может измениться.

Более подробно об этих и многих других настройках вы можете прочитать в документации на официальном сайте: <http://wiki.linux-vserver.org/VServerConfiguration>.

Дмитрий Столяров  
([@mrms-dos.ru](mailto:@mrms-dos.ru))



## «Open Source» приглашает к сотрудничеству!

Электронное приложение «Open Source» всегда открыто для сотрудничества с новыми авторами, с читателями и их конструктивными предложениями по улучшению издания, обоснованной критикой и любыми отзывами, с компаниями, занимающимися разработкой и продвижением программного обеспечения с открытым кодом.

Приветствуются все энтузиасты, желающие опубликовать у нас свои статьи.

Тематика нужных материалов очевидна из предназначения приложения, то есть FOSS (Free and Open Source Software):

теория и практическое применение; исторические сведения, анализ сегодняшнего положения, прогнозы на будущее и другие аспекты, связанные с открытым ПО.

Среди наиболее интересных на данный момент общих тем можно выделить:

- ✓ общие обзоры новых и/или интересных проектов Open Source и конкретных приложений, свежих версий дистрибутивов Linux, \*BSD и других систем;
- ✓ советы и рекомендации новичкам в мире GNU;

- ✓ истории успеха применения/распространения ПО с открытым кодом;
- ✓ философия и идеология Free Software;
- ✓ разработка приложений с применением средств Open Source.

Желательный объем статей: 4800, 9600 или 14400 символов (с пробелами). Но не стоит строго ограничиваться приведенными выше рамками! Если у вас есть свои темы и предложения, присылайте – рассмотрим все вопросы. Не забывайте, что только совместными усилиями мы сможем сделать наше приложение лучше.

Публичное обсуждение «Open Source» проводится в [форуме](#) сайта журнала «Системный администратор».

Связаться с редакцией можно по электронной почте [osa@samag.ru](mailto:osa@samag.ru).

### Подписные индексы:

**20780\***  
**81655\*\***

по каталогу  
агентства  
«Роспечать»

**87836**

по каталогу  
агентства  
«Пресса  
России»

\* **годовой**  
\*\* **полугодовой**

## Подписка на журнал «Системный администратор»

### Российская Федерация

- ✓ Подписной индекс: годовой – **20780**, полугодовой – **81655**  
Каталог агентства «Роспечать»
- ✓ Подписной индекс: **87836**  
Объединенный каталог «Пресса России»  
Адресный каталог «Подписка за рабочим столом»  
Адресный каталог «Библиотечный каталог»
- ✓ Альтернативные подписные агентства:  
агентство «Интер-Почта»  
(495) 500-00-60, курьерская доставка по Москве  
агентство «Вся Пресса»  
(495) 787-34-47  
агентство «Курьер-Пресссервис»  
агентство «ООО Урал-Пресс»  
(343) 375-62-74
- ✓ Подписка On-line  
<http://www.arzi.ru>  
<http://www.gazety.ru>  
<http://www.presscafe.ru>

### СНГ

В странах СНГ подписка принимается в почтовых отделениях по национальным каталогам или по списку номенклатуры «АРЗИ»:

- ✓ **Азербайджан** – по объединенному каталогу российских изданий через предприятие по распространению печати «Гасид» (370102, г. Баку, ул. Джавадхана, 21)

- ✓ **Казахстан** – по каталогу «Российская пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»
- ✓ **Беларусь** – по каталогу изданий стран СНГ через РГО «Белпочта» (220050, г. Минск, пр-т Ф. Скорины, 10)
- ✓ **Узбекистан** – по каталогу «Davriy nashrlar», российские издания через агентство по распространению печати «Davriy nashrlar» (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)
- ✓ **Армения** – по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г. Ереван, пл. Сасунци Давида, д. 2) и ЗАО «Контакт-Мамул» (375002, г. Ереван, ул. Сарьяна, 22)
- ✓ **Грузия** – по списку номенклатуры «АРЗИ» через АО «Сакпресса» (380019, г. Тбилиси, ул. Хошараульская, 29) и АО «Мацне» (380060, г. Тбилиси, пр-т Гамсахурдия, 42)
- ✓ **Молдавия** – по каталогу через ГП «Пошта Молдавей» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)  
по списку через ГУП «Почта Приднестровья» (МД-3300, г. Тирасполь, ул. Ленина, 17)  
по прайс-листу через ООО агентство «Editil Periodice» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
- ✓ Подписка для **Украины**:  
Киевский главпочтамт  
Подписное агентство «KSS»  
Телефон/факс (044)464-0220