

# Веб-Аналитик.ИНФО

[www.web-analitik.info](http://www.web-analitik.info)

## ХОСТИНГ

Диалект Oracle SQL: Model в примерах  
Неприступный почтовый сервер  
Удаленная переустановка Linux  
на примере Debian

## CMS

UMI.CMS: сайт как набор XML-сервисов  
Учимся работать с информационными  
блоками 1С-Битрикс

## ВЕБ-РАЗРАБОТКИ

Специализированный софт:  
купить или написать?  
Требования к html-верстке  
Пишем фильтр «плохих» слов  
Обзор алгоритмов кластеризации  
данных

## ИНТЕРНЕТ

Куда и кому стучит Google Chromium  
Обзор медиаплеера PowerZest HD-500  
Два дня из жизни IT-шника  
Обзор Ubuntu 10.10 Maverick Meerkat  
Alpha 3  
Обновленный файловый менеджер  
Nautilus в Ubuntu 10.10

## ИНТЕРВЬЮ



Интервью с  
Павлом Чебуниным  
Хостинг-компания  
ISPserver



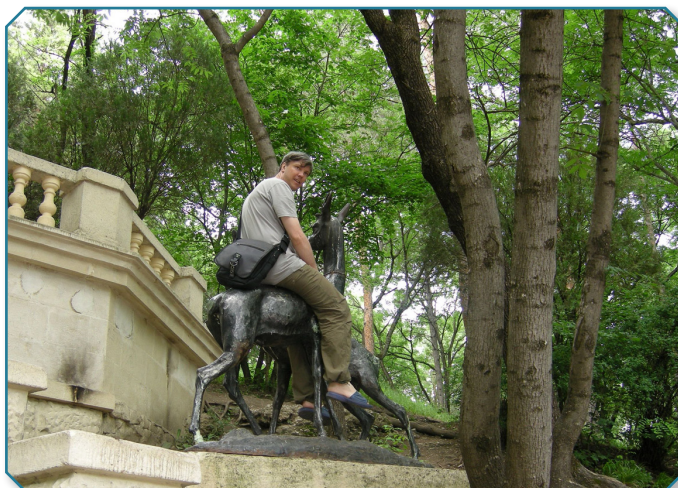
Интервью с  
Тимофеем Горшковым  
Генеральный директор  
компании Инсейлс



Интервью с  
Сергеем Боровиковым  
Руководитель  
Softline Cloud Services



Интервью с  
Дмитрием Агафоновым  
Координатор и руково-  
дитель портала Ubuntu.ru



## Приветствую!

Последние несколько недель, получая новости в своей любимой RSS-читалке, замечая, что борьба за внедрение или не внедрение Свободного Программного Обеспечения (СПО) в образовательных учреждениях набирает свои скрытые обороты. Причем эта борьба идет абсолютно не в том русле, в котором она по идее должна идти. В идеале мы должны бороться за бесплатное и независимое программное обеспечение, а в реале, и «противники» внедрения и внедренцы СПО, прежде всего, рассчитывают именно на материальную выгоду от сего мероприятия. То есть и те и другие в итоге желают «отпилить» N-ое количество бюджетных денег у государства, выделяемых либо на внедрение СПО, либо на продление текущих лицензий. Так по некоторым оценкам, продление текущих лицензий обойдется налогоплательщикам в 240 рублей за одно рабочее место, а внедрение нового СПО по 180 рублей за то же самое одно рабочее место. В одном отчете даже прочитал, что «противники» внедрения СПО приводят некие данные, где на конкретных цифрах показано, что переход на СПО обойдется государству еще дороже, нежели продление текущих лицензий, поскольку учителей (или персонал) придется обучать заново. Честно говоря, до меня никак не дойдет, что такого можно изучать в операционной системе, скажем Ubuntu или Alt Linux, чтобы научиться ей пользоваться? Ведь обычный пользователь ПК, включив компьютер с предустановленной современной Linux системой, разберется с ней ну максимум в течение одного-двух дней. К примеру, мои племянники (одному 10 лет, а другому 12 лет) через три дня знакомства с Ubuntu, уже в легкую ставили сами пакеты и добавляли в систему сторонние репозитории. Современные дистрибутивы Linux (в данном случае отдаю предпочтение Ubuntu) настолько просты и добры к обычному пользователю ПК, что переобучаться там по большому счету просто нечему. Я уже не говорю о том, что непонятно, зачем тратить те самые 180 рублей на внедрение за одно рабочее место, поскольку все свободное ПО лежит в Интернете в свободном доступе, ведь на то оно и свободное ПО. Достаточно скачать образ той же Ubuntu, записать на диск, вставить в дисковод, установить систему (скажем в рамках лабораторной работы) на компьютер, и наслаждаться понятным и удобным интерфейсом, не говоря уже о более чем 40 тысячах (вдумайтесь в эту цифру!) бесплатных программ, доступных из центра загрузки приложений. Все это можно сделать абсолютно бесплатно, без каких-либо дополнительных денежных затрат из нашего и так дырявого государственного бюджета. По некоторым оценкам в России государство тратит более одного миллиарда рублей на лицензионное программное обеспечение. Из них 200 миллионов рублей идут на госзакупки. В очередной раз убеждаюсь, что распил денег налогоплательщиков — это основной мотиватор большинства государственных и около государственных структур.

«Пилите, Шура, пилите... Они золотые...» (с) Паниковский

Станислав Горнаков  
Руководитель проекта Веб-Аналитик.ИНФО

**Хочешь быть в курсе всех событий?**  
**Подпишись на нашу RSS-ленту новостей или Twitter**  
**Мы вещаем круглые сутки семь дней в неделю**



Свежие новости  
в Twitter'e



RSS-лента  
новостей



Журнал выходит ежемесячно  
и распространяется бесплатно

## Издательская группа

ООО «Издательство «Стангор»  
www.stangor.ru  
Журнал Веб-Аналитик.ИНФО  
www.web-analitik.info

Издатель и руководитель проекта  
Станислав Горнаков sg@stangor.ru

Зам. главного редактора  
Игорь Редько ir@stangor.ru

Выпускающий редактор  
Андрей Колисниченко

Арт-директор  
Светлана Петрова  
art-dis@stangor.ru

## Редакторы

Сергей Рубан  
Ирина Войкова  
Андрей Коваль  
Василий Сенявский  
Марат Ягудин  
Игорь Войков  
Игорь Периодов

## Корректоры

Ксения Рубова  
Максим Веров  
Роман Ветрин  
Максим Злобин

## Авторы

Станислав Горнаков  
Данил Костров  
Сергей Мишин  
Лев Хомич  
Алексей Ворошилов  
Виталий Степаненко  
Андрей Часовских  
Алексей Плутенко  
Марат Сибгатуллин  
Алексей Смирнов  
Дмитрий Игнатюк  
Андрей Перминов  
Николай Марков  
Вадим Неворотин  
Олег Данилов  
Алексей Шаферов  
Дмитрий Канн  
Алексей Ужва  
Сергей Кузнецов

## Отдел маркетинга и рекламы

Руководитель отдела  
Виктор Прудников pr@stangor.ru

Реклама в журнале  
siteweb@web-analitik.info

Реклама на сайте журнала  
siteweb@web-analitik.info

Издание зарегистрировано в Комитете  
Российской Федерации по Печати  
Свидетельство № ФС 77 - 39005

## Журнал издает

ООО «Издательство «Стангор»

Для пресс-релизов и информации о  
пресс-конференциях  
siteweb@web-analitik.info

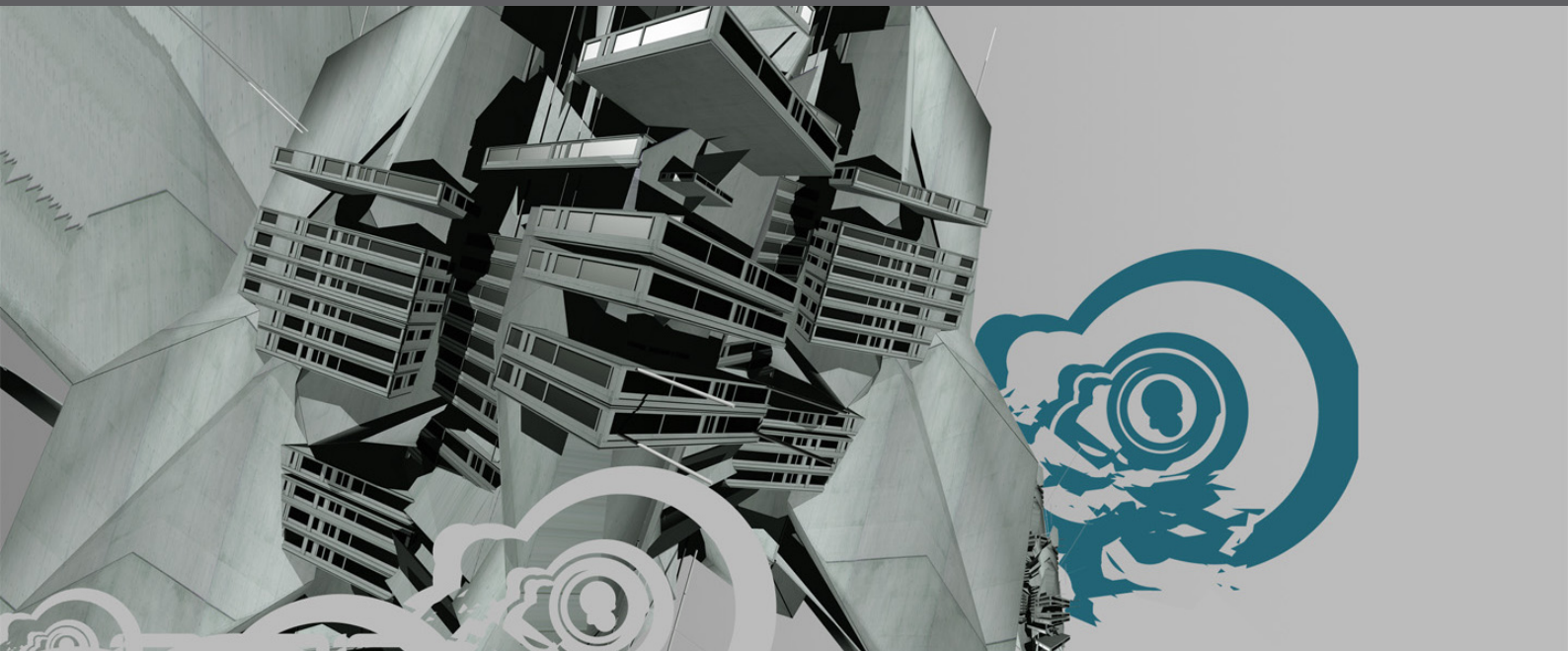
## Авторам

www.web-analitik.info/authors/

За достоверность рекламной информации ответственность несут рекламодатели. Рекламные материалы не редактируются и не корректируются. Редакция ждет ваших откликов и писем читателей. Фотографии, рукописи и другие печатные материалы не редактируются и не корректируются. При цитировании или перепечатывании материалов журнала ссылка на сайт www.web-analitik.info и название журнала Веб-Аналитик.ИНФО обязательна. Полное или частичное воспроизведение материала журналов возможно только с письменного разрешения Издательства Стангор. Мнение редакции журнала может не совпадать с мнением авторов статей публикуемых в журнале. Все товарные знаки принадлежат их владельцам.

© ООО «Издательство «Стангор»  
© WEB-ANALITIK.INFO  
© ВЕБ-АНАЛИТИК.ИНФО





## ХОСТИНГ

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Интервью с Павлом Чебуниным<br>[ Руководитель пресс-службы, хостинг-компания ISPserver ]..... | 06 |
| Диалект Oracle SQL: Model в примерах.....                                                     | 08 |
| Неприступный почтовый сервер, или жизнь без спама.....                                        | 12 |
| Удаленная переустановка Linux на примере Debian.....                                          | 15 |
| Создание полноценного сервера<br>на базе недорогого домашнего роутера.....                    | 18 |

## КАТАЛОГ

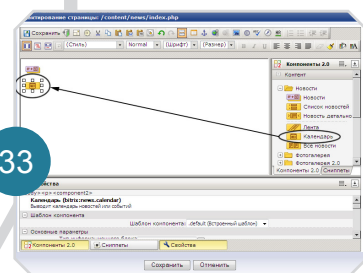
|                       |    |
|-----------------------|----|
| Хостинг-компании..... | 23 |
|-----------------------|----|

## CMS

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| Интервью с Тимофеем Горшковым<br>[ Генеральный директор компании Инсейлс ]..... | 25 |
| UMI.CMS: сайт как набор XML-сервисов.....                                       | 29 |
| Учимся работать с компонентами<br>и информационными блоками CMS 1С-Битрикс..... | 33 |

## КАТАЛОГ

|          |    |
|----------|----|
| CMS..... | 39 |
|----------|----|



## ВЕБ-РАЗРАБОТКИ

- Интервью с Сергеем Боровиковым  
[ Руководитель Softline Cloud Services, компания Softline ]..... **41**
- Простое управление проектами и коммуникациями  
с подрядчиками для заказчиков и владельцев проектов..... **44**
- Специализированный софт: купить или написать?..... **46**
- Требования к html-верстке..... **48**
- Пишем OpenID авторизацию..... **50**
- Пишем фильтр «плохих» слов..... **52**
- Обзор алгоритмов кластеризации данных..... **54**
- Сказ о том, как Bash и SVG спасли от рутины..... **58**

## КАТАЛОГ

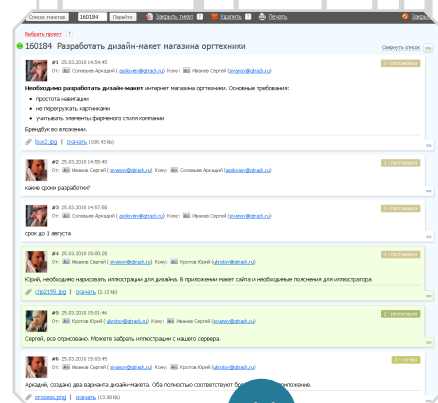
- Веб-студии..... **60**

## ИНТЕРНЕТ

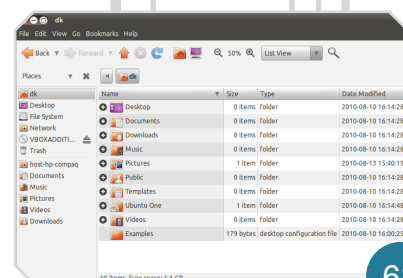
- Интервью с Дмитрием Агафоновым  
[ Координатор и руководитель портала Ubuntu.ru ]..... **62**
- Обзор Ubuntu 10.10 Maverick Meerkat Alpha 3..... **66**
- Обновленный файловый менеджер Nautilus в Ubuntu 10.10..... **68**
- Исследование беспроводного интернета на вокзалах Москвы..... **69**
- Два дня из жизни IT-шника..... **70**

## АВТОРСКАЯ КОЛОНКА

- Куда и кому стучит Google Chromium..... **74**
- Клавиатура с подсветкой..... **76**
- Обзор медиаплеера PowerZest HD-500..... **77**



44



68

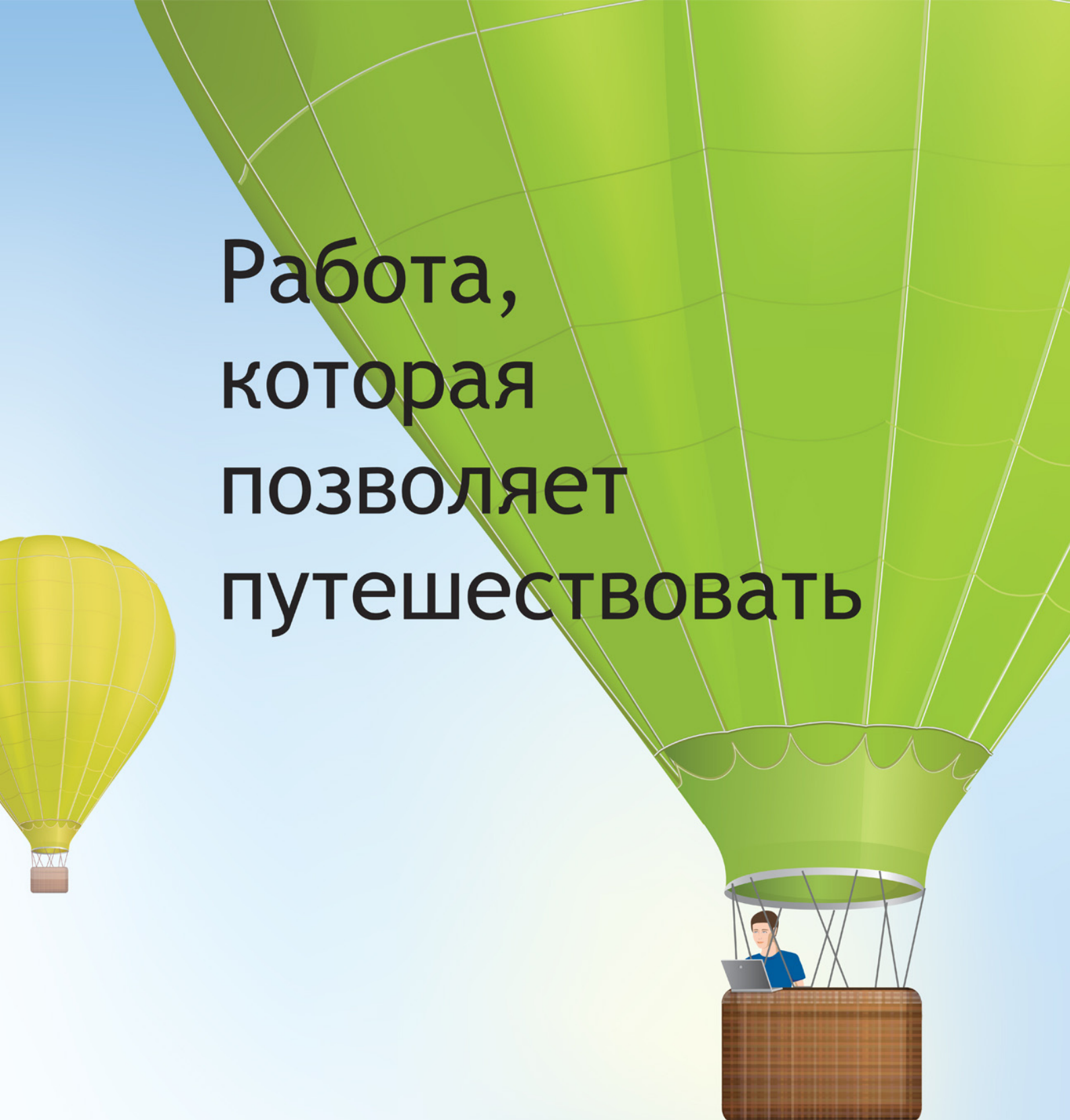


70



77





Работа,  
которая  
позволяет  
путешествовать

 Free-lance.ru

Удаленная работа для дизайнеров,  
программистов, менеджеров,  
оптимизаторов, копирайтеров



## ХОСТИНГ

**Интервью с  
Павлом Чебуниным  
[ Руководитель Пресс-службы  
Хостинг-компания ISPserver  
[www.ispserver.com](http://www.ispserver.com) ]**

**Диалект Oracle SQL: Model  
в примерах**

**Неприступный почтовый  
сервер, или жизнь без спама**

**Удаленная переустановка  
Linux на примере Debian**

**Создание полноценного  
сервера на базе недорогого  
домашнего роутера**





# Интервью с Павлом Чебуниным

**[ Руководитель Пресс-службы  
Хостинг-компания ISPserver  
[www.ispserver.com](http://www.ispserver.com) ]**

*Павел, добрый день! Компания ISPserver недавно запустила пока еще новую для России услугу – облачный хостинг. Расскажите немного, что представляет собой облачный хостинг от ISPserver.*

Да, действительно, в мае этого года мы запустили облачный хостинг. Суть услуги заключается в предоставлении пользователям наших вычислительных мощностей без каких-либо ограничений.

Наш облачный хостинг – это отказоустойчивый виртуальный хостинг с почти неограниченной масштабируемостью. При этом клиенты не «привязаны» к определенным тарифным опциям и не переплачивают за ресурсы, которые не потребляют.

*А какие еще очевидные преимущества у облачного хостинга перед традиционными видами хостинга? Насколько вообще облачные вычисления популярны в мире?*

Зарубежный рынок изобилует предложениями, реализованными по концепции облачных вычислений (cloud computing), ведутся активные диалоги в хостинг-сообществе. Почему? Потому что это, прежде всего, выгодно клиен-

ту. Традиционный виртуальный хостинг предполагает, что на одном физическом сервере размещено несколько веб-проектов разных клиентов, которые в свою очередь делят между собой ресурсы сервера. Данный вид хостинга приемлем для большинства сайтов клиентов, однако, может случиться так, что небольшая информационная страничка начала динамично развиваться, и вот она уже стала высокопосещаемым порталом. Или же, к примеру, компания решила провести акцию у себя на сайте, и в один момент на сайт пришло большое число посетителей. В подобных случаях сайт клиента может потребить большее количество ресурсов физического сервера, чем обычно, и это окажет влияние не только на этот сайт, но и на работоспособность сайтов «соседей», которые также размещены на данном физическом сервере.

Более того, даже если говорить о таком виде хостинга, как аренда выделенного сервера, то внезапный всплеск посещаемости может привести к тому, что не хватит ресурсов и целого выделенного сервера. Преимущество облачного хостинга как раз в том, что сайты клиентов будут доступны в любом случае. Если для клиента критичным является именно работоспособность его веб-проектов, если он заинтересован в эффективном использовании своих денежных средств, то облачный хостинг, станет наилучшим выбором.

*А за счет чего достигается надежность и отказоустойчивость облачного хостинга?*

За счет того, что архитектура облачного хостинга распределена между несколькими физическими серверами с применением отказоустойчивых кластерных технологий. В качестве платформы облачных вычислений мы используем комплекс программного обеспечения ISPmanager Cluster.

*Павел, расскажите подробнее о платформе облачных вычислений ISPmanager Cluster.*

Прежде всего, давайте вспомним, что под кластером мы понимаем объединение однородных элементов в группу, которая функционирует при этом как единое целое. ISPmanager Cluster, как кластерная система, которая распределяет назначенные роли между несколькими серверами и надежно резервирует их, а это значит, что выход из строя одного из серверов никак не повлияет на функционирование системы в целом. Кроме того, веб-интерфейс ISPmanager Cluster спроектирован на основе популярной панели управления ISPmanager Pro, поэтому он очень удобен и привычен. ▶

Облачный  
ХОСТИНГ

ОТ

ISP | SERVER

*Павел, а если я только создал свой сайт, он еще нигде не «раскручен», то нужно ли мне платить за такой сайт, который по сути пока простаивает?*

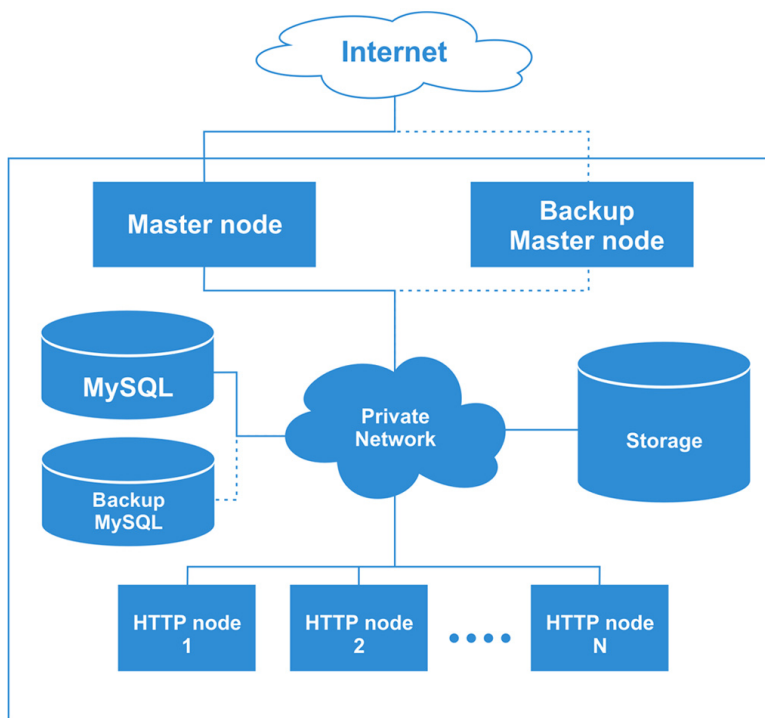
В данном случае вы будете платить только небольшую абонентскую плату за использование дискового пространства и все. Это действительно очень удобно и выгодно как для начинающих стартапов, так и для компаний, которые находятся на стадии контентного наполнения своего корпоративного сайта. Зачастую бывает, что проходит достаточно длительный срок между регистрацией домена, размещением сайта на хостинге и его реальной эксплуатацией, тем более отдачей от него. То есть, конверсия наступает не сразу, а между тем на любом хостинге вам сразу нужно полностью оплачивать размещение сайта по выбранному тарифному плану. Поэтому на облачном хостинге экономия для таких сайтов очевидна.

*Все это звучит впечатляюще, сразу захотелось попробовать...*

Нет ничего проще. Наш облачный хостинг можно попробовать бесплатно в течение 5 тестовых дней и самостоятельно оценить для себя все его преимущества.

*А скажите, Павел, какие услуги еще предоставляет компания ISP-server? Как давно работает на рынке?*

Наша компания работает на рынке хостинга с 1997 года. Причем изначально мы были ориентированы на зарубежный рынок. Сейчас мы предоставляем услуги хостинга по всему миру. Дата-центры, в которых размещено наше собственное оборудование, расположены на территории РФ, США и Европы. В свое время компания ISPserver стояла у истоков возникновения в России тогда еще нового веяния в области хостинга – технологии виртуальных выделенных серверов (VDS/VPS). На западе этот вид хостинга называют VPS. Мы же изначально стали именовать данный вид хостинга VDS и в России вошла в обиход именно эта аббревиатура. На текущий момент, мы предоставляем традиционный пере-



**Распределение ролей в ISPmanager Cluster**

чень услуг хостинг-провайдера. Однако если говорить о преимуществах для наших клиентов, то это, прежде всего, бесплатное предоставление программного обеспечения для управления веб-хостингом от ведущего разработчика компании ISPsystem. Ну, и, безусловно, нашим «коньком» всегда была и остается круглосуточная многоязыковая служба технической поддержки. Любой наш клиент из любой точки мира и в любое время может обратиться за помощью и в оперативном режиме получить исчерпывающий ответ.

*Расскажите немного о своих клиентах. На кого вы, прежде всего, ориентированы?*

Мы всегда позиционировали себя как надежная компания, репутация которой подтверждена многолетним успешным опытом работы на рынке. География продаж наших услуг рассредоточена по всему миру. Мы ориентированы на клиента, который тщательно подходит к выбору поставщика услуг. Клиента, который ценит качество, надежность и комплексный подход, который привык экономить свое время и деньги.

*Многие хостинг-провайдеры практикуют партнерские программы, предусмотрены ли они у вас?*

Да, безусловно, для всех предоставляемых видов хостинга у нас предусмотрена партнерская программа. Это значит, что абсолютно любой наш клиент может стать нашим партнером и зарабатывать вместе с нами. На сайте регулярно обновляются рекламные материалы для маркетинговой поддержки наших партнеров, также мы всегда открыты для любых предложений о совместном сотрудничестве.

*Павел, расскажите, а что в планах?*

Мы планируем развиваться в сторону SaaS провайдинга.

На сегодняшний момент достигнут ряд договоренностей с разработчиками различных бизнес-приложений, ведется методологическая, организационная и техническая работа по запуску услуг по модели SaaS. Кроме того, для нас уже стало доброй традицией постоянно совершенствовать услуги для существующих клиентов. Мы постоянно работаем над модернизацией нашего оборудования, что позволяет нам регулярно увеличивать количество ресурсов на текущих тарифных планах. В итоге наши клиенты получают больше за те же деньги.

*Спасибо за интервью, желаем компании ISPserver реализации намеченных планов.*

**ISP | SERVER**



Автор

Лев Хомич  
www.exaide.ru

# Диалект Oracle SQL: Model в примерах



Любой разработчик имеющий дело с генерацией отчетности из баз данных, регулярно сталкивается с построением громоздких запросов. Часто это бывает связано с ошибками проектирования БД, и еще чаще со сложностями преобразования извлекаемых данных. К последним можно отнести применение итерационных методов вычисления, подсчет промежуточных итогов по подгруппам, расчеты, в которых используются значения соседних строк выборки, сложное форматирование строк и подобные задачи. Такие преобразования часто выносятся с уровня БД на уровень сервера приложений или клиента, что сказывается на производительности и удобстве сопровождения кода. Для решения этих задач SQL СУБД Oracle предоставляет аналитические функции и оператор `MODEL`, о нем и пойдет речь в этой статье.

Расширение конструкции `SELECT` стало доступно с СУБД версии 10g. `MODEL` позволяет обращаться к данным выборки как к многомерным массивам, изменять и добавлять элементы, проводить сложную агрегацию, а также решать ряд задач, которые до этого требовали использования PL/SQL. При этом языковые конструкции остаются читабельными и декларативными. Одним словом – Excel-like, плюс вся нагрузка ложится на плечи сервера БД.

## Синтаксис

```
MODEL [IGNORE NAV] [RETURN UPDATED ROWS]
[PARTITION BY (partition_column_1, ...)]
DIMENSION BY (dimension_column_1, ...)
MEASURES (measured_column_1, ...)
RULES [AUTOMATIC ORDER | ITERATE (value)
[UNTIL (expression)]] (
    rule_1, ...
);
```

Оператор `MODEL` обрабатывается в числе последних, после него только `DISTINCT` и `ORDER BY`. В результате применения столбцы выборки отображаются в массивы `measured_column_*` с измерениями `dimension_column_*`. Обязательный параметр `PARTITION BY` определяет паритиции, аналогичные используемым в аналитических функциях (каждая из них обрабатывается правилами `rule_*` как независимый массив). Правила применяются в порядке перечисления.

## Простейшие примеры

Для начала смоделируем выборку чисел 1, 2 и 3:

```
SELECT *
FROM dual
MODEL DIMENSION BY (0 dimension)
MEASURES (dummy)
RULES (
    dummy[5] = 1,
    dummy[6] = 2,
    dummy[7] = 3
);
```

| DIMENSION | R |
|-----------|---|
| 0         | X |
| 7         | 3 |
| 6         | 2 |
| 5         | 1 |

В данном случае, по трем правилам заполняется массив `dummy` с измерением `dimension`. Алиас `0 dimension` определяется для того, чтобы добавить новый столбец. Разберем преобразование подробнее. Первым делом происходит определение и отображение столбцов выборки (`0 dimension` в `DIMENSION`, `dummy` в `MEASURES`), затем по этим столбцам происходит выборка (возвращается строка `dummy = X`, `dimension = 0`) и только после этого выполняются правила. Сначала ищется строка с `dimension = 5`, но т.к. она не находится, создается новая и заполняется `dummy = 1`, аналогично для оставшихся двух правил. Если необходимо, с помощью директивы `RETURN UPDATED ROWS` можно вывести только обновленные строки:

```
SELECT result, dummy
FROM dual
MODEL RETURN UPDATED ROWS
DIMENSION BY (dummy)
MEASURES (0 result)
RULES (
    result[5] = 1, ▶
```

```
result[6] = 2,
result[7] = 3
);
```

| RESULT | DUMMY |
|--------|-------|
| 3      | 7     |
| 2      | 6     |
| 1      | 5     |

Применять наборы правил можно и в циклах. Следующий запрос рассчитывает несколько элементов последовательности Фибоначчи:

```
SELECT sequence
FROM dual
MODEL DIMENSION BY (0 dimension)
MEASURES (0 sequence)
RULES ITERATE (100500) UNTIL (sequence[iteration_number] > 10) (
sequence[iteration_number] =
CASE iteration_number
WHEN 0 THEN 0
WHEN 1 THEN 1
ELSE sequence[iteration_number - 2]
+ sequence[iteration_number - 1]
END
);
```

| SEQUENCE |
|----------|
| 0        |
| 1        |
| 1        |
| 2        |
| 3        |
| 5        |
| 8        |
| 13       |

ITERATE задает количество итераций цикла (начиная с 0), а необязательная директива UNTIL – условие выхода из него (которое сработало, судя по тому что вы еще не нажали Ctrl + End). Доступ к счетчику осуществляется через переменную iteration\_number.

## Диапазоны и агрегирование

Выше мы видели примеры с расчетом значений отдельных элементов. Кроме этого, правила можно задавать для их групп, отфильтрованных по заданным условиям. Для этого используется второй тип ссылок – символические (ранее были только позиционные). Индексы в символических ссылках могут содержать проверку условий истинности, например:

- cnt[day < 6, type LIKE 'latt%']
- cnt[day IN (3, 6), cv(type)]
- cnt[day BETWEEN 1 AND 16, regexp\_like(type, '^(.+)(sso|tte)\$')]

В позиционных ссылках значения индексов определены однозначно:

- cnt[2, 'black']
- cnt[7, 'latte']

С помощью символических ссылок нельзя создавать новые элементы, а только обновлять существующие (если очень хочется, то, конечно, можно, но об этом далее). Позиционные же позволяют и обновлять, и создавать. С диапазо-

нами, заданными с помощью символических ссылок допустимо использовать любые агрегатные функции. Аналитические функции внутри правил запрещены.

Рассмотрим таблицу, хранящую информацию о выпитом в течение недели кофе:

```
SELECT * FROM coffee;
```

| TYPE     | CNT | DAY |
|----------|-----|-----|
| turkish  | 1   | 1   |
| espresso | 1   | 1   |
| turkish  | 2   | 2   |
| black    | 1   | 2   |
| espresso | 1   | 2   |
| latte    | 3   | 3   |
| black    | 2   | 4   |
| ice      | 1   | 4   |

Предположим, мы хотим вывести отчет о том, сколько всего чашек кофе выпито в каждый из дней и заодно учтем, что в четверг порции черного кофе были двойными. Итак, запрос:

```
SELECT *
FROM coffee
MODEL DIMENSION BY (day, type)
MEASURES (cnt)
RULES (
cnt[4, 'black'] = cnt[cv(day), 'black'] * 2,
cnt[FOR day FROM 1 TO 4 INCREMENT 1, '
total for day'] = sum<(cnt)[cv(day), ANY],
cnt[NULL, 'GRAND TOTAL'] = sum(cnt)[ANY,
'total for day']
)
ORDER BY day, type DESC;
```

| DAY | TYPE          | CNT |
|-----|---------------|-----|
| 1   | turkish       | 1   |
| 1   | espresso      | 1   |
| 1   | total for day | 2   |
| 2   | turkish       | 2   |
| 2   | espresso      | 1   |
| 2   | black         | 1   |
| 2   | total for day | 4   |
| 3   | latte         | 3   |
| 3   | total for day | 3   |
| 4   | ice           | 1   |
| 4   | black         | 4   |
| 4   | total for day | 5   |
|     | GRAND TOTAL   | 14  |

Разберем правила подробнее. Первое удваивает количество выпитого в четверг кофе. Функция cv(dimension\_name) возвращает текущее значение индекса по измерению dimension\_name для обрабатываемого элемента (т. е. в данном случае, вместо cv(day) можно было указать day = 4 или, при желании, сослаться на предыдущий день как day = cv(day) - 1). Второе правило вычисляет подрезультаты с понедельника по четверг. Кроме цикла (который и так нагляден), здесь используются ссылки на диапазоны элементов в правых частях равенств. Указывать диапазоны можно теми же способами, что и проверки в конструкции WHERE, дополнительное ключевое слово ANY служит для выбора любых значений индекса. Ссылки на диапазоны в правой части равенства необходимо агрегировать, в данном случае используется функция sum. И, наконец, третье правило считает сумму подрезультатов.

Двигаемся дальше. Рассмотрим запрос, выводящий подрезультаты выпитого по видам кофе, количество латте, выпитого во вторник, а также сообщение о том, достигнута ли цель недели – выпить эспрессо в среду: ▶

```

SELECT *
FROM coffee
MODEL DIMENSION BY (day, type)
MEASURES (cnt)
RULES (
  cnt[NULL, FOR type IN (SELECT DISTINCT
type FROM coffee)] = sum(cnt)[ANY, cv(type)],
  cnt[NULL, 'GRAND TOTAL'] = sum(cnt)[NULL,
ANY],
  cnt[NULL, 'drank ' || cnt[2, 'latte'] ||
' cups of latte on 2 day'] = NULL,
  cnt[NULL, CASE
    WHEN cnt[3, 'espresso'] IS PRESENT
THEN ' ACHIEVED'
    ELSE ' FAILED'
  END || ': drank espresso on 3 day'] = NULL
)
ORDER BY day, type DESC;

```

| DAY | TYPE                            | CNT |
|-----|---------------------------------|-----|
| 1   | turkish                         | 1   |
| 1   | espresso                        | 1   |
| 2   | turkish                         | 2   |
| 2   | espresso                        | 1   |
| 2   | black                           | 1   |
| 3   | latte                           | 3   |
| 4   | ice                             | 1   |
| 4   | black                           | 2   |
|     | turkish                         | 3   |
|     | latte                           | 3   |
|     | ice                             | 1   |
|     | espresso                        | 2   |
|     | black                           | 3   |
|     | GRAND TOTAL                     | 12  |
|     | drank cups of latte on 2 day    |     |
|     | FAILED: drank espresso on 3 day |     |

В первом правиле используется цикл с итерациями по значениям выборки. Вложенные запросы, используемые внутри MODEL должны быть некоррелированными. Следующее правило вычисляет итоговую сумму. Обратите внимание на строку drank cups of latte on 2 day. Так как элемент cnt[2, 'latte'] не был найден, мы получили NULL по ссылке. Это поведение можно изменить директивой IGNORE NAV (добавляется после слова MODEL), тогда вместо ненайденных элементов и NULL, в расчеты будут подставляться: 0 – для чисел, 1 января 2001 года – для дат, пустая строка – для строковых типов и NULL для всего остального. И, наконец, четвертое правило демонстрирует использование выражения IS PRESENT, оно возвращает истину в случае, если заданный элемент существует, но, увы, эспрессо в среду не было выпито.

Применение конструкции MODEL запрещает использовать агрегатные функции внутри других блоков SELECT (вообще, все выбираемые столбцы должны быть производными от упоминаемых в MODEL). Вместо этого следует объявлять эти функции внутри PARTITION BY, DIMENSION BY или MEASURES. Например, следующему простому запросу

```

SELECT employee_id, sum(amount)
FROM sales
GROUP BY employee_id;

будет эквивалентен

SELECT employee_id, amt
FROM sales
GROUP BY employee_id

```

```

MODEL PARTITION BY (employee_id)
DIMENSION BY (0 dummy)
MEASURES (sum(amount) amt)
RULES ();

```

Для контроля изменений, вносимых правилами, существуют три семантики: UPDATE, UPSERT, UPSERT ALL. UPDATE разрешает только обновлять элементы, UPSERT (работает по умолчанию) изменять и добавлять с помощью позиционных ссылок, а UPSERT ALL разрешает создавать элементы, используя символические ссылки. Изменять семантику можно как для индивидуального правила (в этом случае директива записывается перед ним), так и на уровне всего блока (тогда директиву нужно указать после RULES).

Важно знать, что UPSERT ALL работает не по аналогии с циклом FOR. Создание новых элементов в этом случае происходит в четыре шага.

1. Находятся все элементы, которые соответствуют символической ссылке.
2. По каждому измерению находится набор уникальных значений индексов.
3. Вычисляется Декартово произведение этих множеств.
4. Все несуществующие элементы, присутствующие в произведении, создаются.

## Сложные примеры

В завершении пара примеров, которые покажут насколько разные задачи могут решаться с помощью MODEL. Для начала выведем список выпитого кофе для каждого дня одной строкой:

```

SELECT day, substr(type, 2) listing
FROM coffee
MODEL RETURN UPDATED ROWS
PARTITION BY (day)
DIMENSION BY (row_number() OVER (PARTITION
BY day ORDER BY type) position)
MEASURES (type, cnt)
RULES ITERATE (100500) UNTIL (presen-
tv(type[iteration_number + 2], 1, 0) = 0) (
  type[0] = type[0] || ' ' || cnt[iteration_
number + 1] || ' x ' || type[iteration_number +
1]
)
ORDER BY day; ►

```





| DAY | LISTING                              |
|-----|--------------------------------------|
| 1   | 1 x espresso, 1 x turkish            |
| 2   | 1 x black, 1 x espresso, 2 x turkish |
| 3   | 3 x latte                            |
| 4   | 2 x black, 1 x ice                   |

Из нового здесь функция `presentv(ref, arg1, arg2)`, она возвращает `arg1`, если ссылка `ref` указывает на существующий элемент с `NOT NULL` значением, и `arg2` в противном случае. Для нумерации строк и сортировки по названию используется аналитическая функция (`row_number() OVER (PARTITION BY day ORDER BY type)`). Обработка отдельно для каждого дня обеспечивается партиционированием (`PARTITION BY`) как в модели, так и в аналитической функции.

И, в завершении, геометрический пример. На прямой дан набор пересекающихся отрезков [`p1`, `p2`], требуется вывести отсортированный список диапазонов, целиком покрытых отрезками:

```
SELECT * FROM lines;
```

| P1 | P2 |
|----|----|
| 1  | 6  |
| 5  | 7  |
| 2  | 4  |
| 8  | 20 |
| 11 | 28 |
| 30 | 32 |

Считается, что `p2 >= p1`, решение для произвольно хранящихся отрезков оставляется для разминки заинтересовавшимся. Итак, код примера:

```
SELECT p1, max(p2) p2
```

```
FROM (
```

```
SELECT p1, p2 FROM lines
```

```
MODEL DIMENSION BY (row_number() OVER (ORDER BY p1, p2) rn)
```

```
MEASURES (p1, p2)
```

```
RULES AUTOMATIC ORDER (
```

```
p1[rn > 1] ORDER BY rn = CASE sign
(p2[cv(rn) - 1] - p1[cv(rn)])
```

```
WHEN 1 THEN p1[cv(rn) - 1]
```

```
ELSE p1[cv(rn)]
```

```
END,
```

```
p2[rn > 1] ORDER BY rn = greatest (p2[cv(rn) - 1], p2[cv(rn)])
```

```
)
```

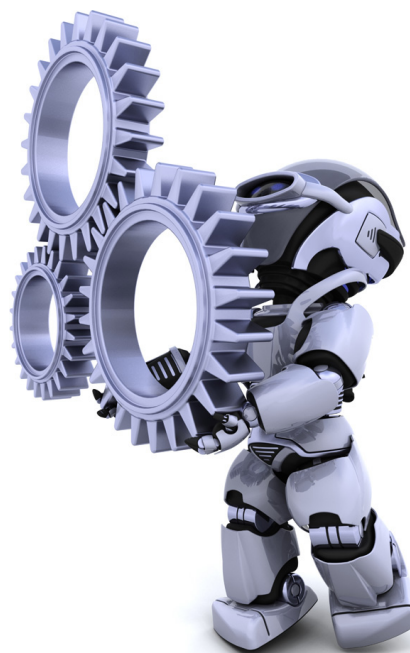
```
)
```

```
GROUP BY p1
```

```
ORDER BY p1;
```

| P1 | P2 |
|----|----|
| 1  | 7  |
| 8  | 28 |
| 30 | 32 |

Отрезки в модели сначала сортируются по левым точкам и нумеруются (`row_number() OVER (ORDER BY p1, p2)`), а затем просматриваются сверху вниз по номеру (директива `ORDER BY rn`). Обратите внимание, что в левой стороне равенства впервые используется символическая ссылка, именно поэтому необходимо указать порядок, в котором будут обрабатываться элементы. Если точка `p1` просматриваемого отрезка принадлежит предыдущему отрезку, то она заменяется на `p1` предыдущего отрезка. `p2` заменяется на самые правые `p2` среди текущего и предыдущего отрезков. Таким



образом, спускаясь вниз, мы расширяем диапазон, перекрытый отрезками. Директива `AUTOMATIC ORDER` используется для того, чтобы `p1` и `p2` считались по очереди для текущего отрезка. Если убрать директиву, сначала будет выполнено первое правило для всех строк, и только затем второе. В общем случае, данная директива учитывает зависимости между элементами при обработке правил и может нарушать порядок их обхода (о чем было написано выше).

## Практическое применение и производительность

Вообще, `MODEL` применяется достаточно редко. Происходит так потому, что 90% действительно сложных вещей решаются аналитическими функциями, а те 10%, которые остаются для `MODEL` традиционно выносятся с уровня БД. Несмотря на это, вот основные сценарии применения:

- необходимо сгенерировать действительно сложный отчет (читабельность и удобство сопровождения приоритетны);
- выборка неосуществима даже с использованием аналитических функций;
- запрос динамически генерируется на высокоуровневом языке (проще сгенерировать правила в `MODEL`, чем аналитику);
- используются ресурсоемкие итерационные вычисления, когда `MODEL` еще хватает, а переход `PL/SQL` ведет к падению производительности;
- академический интерес (декларативный подход вместо процедурного).

С производительностью все просто. `MODEL` в подавляющем большинстве случаев быстрее аналогичного `PL/SQL` кода, но медленнее, чем использование аналитических функций. С помощью `EXPLAIN PLAN` можно узнать некоторые подробности об обработке запроса.

- `SQL MODEL ORDERED [FAST]` — означает, что обработка правил идет в последовательности, заданной в запросе (по умолчанию);
- `SQL MODEL ACYCLIC [FAST]` — автоматически просчитываются зависимости ячеек;
- `SQL MODEL CYCLIC` — самый медленный вариант, при сложной зависимости между ячейками.

Наличие аннотации `FAST` означает, что все просто отлично. Так бывает при соблюдении условия, что все левые части равенств — позиционные ссылки, либо в левой части символические ссылки, но при этом в правой — простые арифметические агрегатные функции (`sum`, `avg` и т. п.). Соответственно, `MODEL FAST` по быстродействию близок к использованию аналитических функций, а `MODEL CYCLIC` может проигрывать реализации на `PL/SQL`. ■



Автор

Вадим Неворотин  
г. Санкт-Петербург  
www.help.ubuntu.ru

# Неприступный почтовый сервер, или жизнь без спама

**Б**орьба со спамом — это головная боль всех ответственных администраторов почты. Чего только они не изобретают, чтобы пользователям лучше жилось. Однако, как показала практика общения со многими системными администраторами, почему-то далеко не все представляют как правильно фильтровать спам.

Чаще всего встречается подход «добавим кучу RBL (DNSBL) и будем радоваться жизни», который практически полностью не верный. Второй по популярности подход — это контент-фильтры, зачастую купленные за бешенные деньги. Такой подход тоже в большинстве случаев совершенно не оправдан.



*Статья написана с прицелом на настройку почтового сервера Postfix, но в целом носит скорей теоретический характер. Описанные опции Postfix нужно указывать в соответствующих \*\_restriction параметрах конфигурационного файла.*

## Немного об SMTP протоколе

Электронная почта имеет много аналогий с обычной почтой. Для нас самое главное сейчас то, что вся информация на электронном «конверте» представляет собой всего лишь два адреса: получателя и отправителя, а также штампик почтальона, и конверт доставившего.

Немного отвлечемся: представьте, что к вам придет лицо крайне отталкивающей наружности и вручит плотно запечатанную посылку с обратным адресом «Трям из Тилимилитрямдии». Рискнете принять и открыть? Вряд ли. Так вот, электронную почту можно тоже легко проверять и отсеивать исходя только из адресной информации, при-

чем простор для возможных действий тут гораздо шире.

Как вам должно быть известно, почта в интернете передается между почтовыми серверами по протоколу SMTP. Любое общение по этому протоколу начинается с трех обязательных заголовков: HELO, MAIL FROM и RCPT TO. То есть перед тем, как начать передавать какие-либо данные, сервер сначала представляется (HELO), потом сообщает обратный адрес отправителя (MAIL FROM) и затем адрес получателя (RCPT TO). Эти три заголовка и есть подпись на электронном конверте, и практически весь спам можно отсеять только исходя из их анализа. Большинство попыток передать что-то моему серверу не доходят дальше MAIL FROM, то есть письма отсеиваются еще до фактического принятия, что значительно снижает нагрузку. То есть вместо того, чтобы открыть посылку от Тряма и обнаружить там споры сибирской язвы, я сразу посылаю почтальона куда подальше.

Итак, что же надо делать, чтобы не принимать письма, заведомо являющиеся спамом? Пойдем по порядку.

## Немного о DNS

Когда-то на заре Интернета почта доставлялась непосредственно на узлы, указанные в почтовом адресе. То есть для доставки письма для user@domain.com почтовый сервер искал IP адрес domain.com и пытался послать посылочку по найденному IP. Потом появились MX записи, которые разом решили большинство проблем подобной организации почтового взаимодействия. Однако некоторые программы все еще могут работать с A-записями при доставке почты. Но у вас, конечно, есть хотя бы одна MX запись для вашего домена, не так ли?

MX записи содержат адреса серверов, на которые должны доставляться письма для указанного домена. Однако в целях борьбы со спамом появилась технология, позволяющая указывать в

DNS также адреса серверов, с которых могут приходить письма от указанного домена. Имя ей Sender Policy Framework.

Подробно вдаваться во все тонкости технологии я не буду, скажу лишь, что TXT запись

```
v=spf1 +mx -all
```

для вашего домена сообщит всем клиентам, поддерживающим проверку SPF, что письма из вашего домена могут приходить исключительно с серверов, указанных в MX записях. Можно сделать правило мягче, написав ~all вместо -all.

Всегда прописывайте SPF запись для домена, а так же включайте проверку SPF на своих почтовых серверах. Я рекомендую жестко запрещать отправку писем из вашего домена со всех хостов, кроме ваших MX серверов. Вкупе с проверкой SPF на вашем сервере, подобная настройка сразу срежет все письма, посылаемые со сторонних хостов от имени пользователей вашего домена на адреса пользователей вашего же домена. А такого спама чуть ли не половина, поскольку обычно SMTP серверы очень плохо защищены от писем из своего же собственного домена, и спамеры этим активно пользуются. SPF раз и навсегда избавит вас от писем Васе Пупкину, написанных судя по конверту Васей же Пупкиным, но пришедших с сервера в каком-нибудь Никарагуа. Как настроить SPF в Postfix вы можете узнать сами, так что не будем тратить время на технические подробности.

Есть еще пару крайне важных замечаний по поводу DNS. Скорее всего, вы знаете, что основные записи DNS, так называемые A-записи, преобразуют имя в IP адрес. Кроме них есть еще CNAME записи, которые назначают псевдоним уже существующему имени. Именно эти два типа записей составляют основу всей системы доменных имен.

Но мало кто из пользователей знает, что есть так же обратные записи, кото-

рые преобразуют IP в доменное имя. Они носят название PTR. Так вот, есть два написанных (строго говоря) правила, которым все же все следуют:



**Для каждой А-записи должна существовать зеркальная PTR запись, то есть по имени хоста через DNS получаем IP, а по IP – обратно то же имя хоста.**



**В качестве адреса в MX записи всегда должно стоять имя (не IP!) хоста, для которого существует А-запись. То есть нельзя, чтобы в MX записи стоял IP или псевдоним (CNAME).**

Если вы не соблюдете одно из этих требований, то приготовьтесь к тому, что минимум четверть почты с вашего домена будет распознаваться как спам. Обусловлено это простым тезисом: благонадежный отправитель всегда все настраивает соблюдая правила, соответственно если правила не соблюдены, то отправителю доверять не следует, а значит и принимать от него почту тоже.

Ну а чтобы включить проверку PTR у себя, используйте опцию

```
reject_unknown_client_hostname
```

Она требует, чтобы IP, с которого совершается соединение, резолвился в имя через PTR, а это имя резолвилось в свою очередь обратно в искомый IP.

Есть и менее жесткое ограничение, задаваемое опцией

```
reject_unknown_reverse_client_hostname
```

В этом случае сервер будет проверять только наличие PTR записи, но не будет требовать существования соответствующей А-записи.

## Проверяем приветствие

Итак, кто-то захотел передать вашему серверу письмо. Передача начинается с приветствия – заголовка HELO. В HELO должно быть указано полное доменное имя (FQDN) отправителя, соответственно если это не так, то можете смело сразу же отказывать в принятии. В Postfix для этого служат две опции:

```
reject_invalid_helo_hostname
```

```
reject_non_fqdn_helo_hostname
```

Первая запрещает прием писем от хостов, передающих приветствие с некорректным синтаксисом, вторая – от хостов, передающих не FQDN в HELO запросе. Однако не FQDN передают только самые глупые спамеры (и про-

дукты MS, но им, как известно, законы не писаны), в конце концов представиться gmail.com не составляет труда. Поэтому надо еще немного присмотреться к HELO. Для этого служит опция

```
reject_unknown_helo_hostname
```

запрещающая прием писем от серверов, представляющих адресом, для которого не существует А- или MX записи.

## Отправитель - стоит ли ему доверять?

Итак, сервер успешно вам представился, следующим пунктом программы идет адрес отправителя. Из него тоже можно извлечь массу полезной информации. Сразу хочу заметить, что адрес отправителя вовсе не обязан быть из того же домена, что и сам сервер. Это распространенное заблуждение, поэтому имейте ввиду, что это не так. Один почтовый сервер спокойно может обслуживать несколько доменов.

Однако если в адресе отправителя указан домен, который вовсе не существует, то письмо совершенно очевидно принимать не стоит. И уж точно не стоит принимать письмо, в котором в качестве обратного адреса указано нечто, вообще адресом не являющееся. За отказ в принятии для таких писем отвечают две опции:

```
reject_non_fqdn_sender
```

```
reject_unknown_sender_domain
```

Первая – проверка адреса на написание, вторая – проверка существования домена.

Уже неплохо, но можно сделать кое-что еще. Можно запросить сервер, обслуживающий указанный адрес отправителя, на предмет существования на нем пользователя с этим адресом. Действительно, вроде бы неплохая идея удостовериться в том, что обратный адрес действительно существует, иначе нам вполне может придти письмо от

эфемерного фантома, о котором никто и не слышал.

Технически это реализуется очень просто: наш сервер открывает встречную SMTP сессию, пытаясь послать письмо по адресу отправителя. Если удастся успешно пройти этап послышки RCPT TO с этим адресом, т. е. если принимающий сервер не заявляет, что указанного ящика на нем нет, то считается, что присланный нам обратный адрес существует. Данные (то есть письмо) при проверке естественно никакие не передаются, сессия прерывается после RCPT TO.

За такую проверку обратного адреса отвечает опция

```
reject_unverified_sender
```

Из сказанного выше следует, что для любого адреса, с которого вы рассылаете почту из своего домена, должен существовать ящик на вашем сервере. Иначе ваши письма не пройдут проверку обратного адреса на стороне получателя и соответственно не будут доставлены по назначению. Это актуально для всяких рассылок и прочей, вроде как односторонней коммуникации, не требующей ответа. Всегда создавайте ящики для всех адресов, с которых вы рассылаете почту. Если вы не хотите получать ответы на какой-то адрес, то просто посылайте письма на него в /dev/null, но принимать эти письма вы обязаны.

## А получатель-то вообще существует?

Вот мы и дошли до последнего заголовка конверта – до получателя. Тут все просто: во-первых, хорошо бы проверить, что переданная нам информация является адресом электронной почты. Для этого служит директива

```
reject_non_fqdn_recipient
```

Кроме того нам бы не хотелось принимать почту на адреса, для которых у нас нет почтовых ящиков. Чтобы на-





строить такое поведение надо сначала создать список обслуживаемых ящиков, дальше рассказать о нем Postfix через один из `*_recipient_maps` параметров конфигурационного файла, ну а дальше либо воспользоваться параметром конфигурационного файла

```
smtpd_reject_unlisted_recipient = yes
```

Либо запрещающей опцией, имеющей тот же эффект:

```
reject_unlisted_recipient
```

В любом случае Postfix перестанет принимать письма для обслуживаемых доменов, если не существует ящика для получателя. Однако это ограничение никак не скажется на пересылке корреспонденции на адреса, которые не находятся в обслуживаемых доменах.

Ну и наконец можно вообще запретить открытую пересылку писем через ваш Postfix, оставив только возможность принимать письма на известные адреса. Для этих целей служит опция

```
reject_unauth_destination
```

Она запрещает отсылку писем всем незарегистрированным пользователям (да, вам придется настроить SMTP авторизацию). Всегда используйте эту опцию! Иначе быстро попадете во всякие DNSBL.



*Вот так только на основе анализа трех заголовков конверта можно отсеять огромное количество спама. Однако спамеры хитрые, поэтому этого все же недостаточно.*

## Грейлистинг

Иногда почтовые серверы бывают перегружены и не могут принять письмо. Как вы думаете, что они отвечают на входящие запросы в этом случае? Как ни странно, так и отвечают — сервер временно недоступен, попробуйте позже. Ни один нормальный отправитель никогда в этом случае не посчитает, что письмо доставить нельзя со всеми вытекающими последствиями. Напротив, отправитель предпримет попытки доставить письмо позже, поставив его в свою очередь на отправку. Этот факт можно (и без сомнения нужно!) очень эффективно использовать: при каждой первой попытке соединения с незнакомым хостом, наш сервер будет отправлять сообщение о временной ошибке, а пропускать письмо только со второго раза. Это отсеет сразу чуть ли не весь оставшийся спам, поскольку спамсерверы практически никогда не предпринимают больше одной попытки доставки письма (иначе они бы просто «упали» от переполнения очереди). Эта технология называется Greylisting, и использовать ее в современных реалиях просто необходимо.

Минусом применения является небольшая (обычно не более получаса) задержка в доставке писем при первом соединении от неизвестного хоста. То есть если еще не известный нашему постфиксу сервер хочет передать письмо, то при первой попытке соединения ему отправляется ошибка о временной недоступности. Если сервер повторил попытку соединения, то письмо принимается, а сервер заносится в надежные узлы и в дальнейшем письма от него принимаются без задержек.

## Блоклисты, или как делать не надо

Некоторые администраторы почты при фильтрации спама полагаются на так называемые DNSBL (RBL) — черные списки узлов, замеченных в рассылке спама. Так вот, никогда не добавляйте никаких проверок DNSBL на ваши почтовые сервера. Тому есть две причины: первая, и самая основная, кроется во второй части первого предложения этого раздела. В эти списки узлы заносятся совершенно беспорядочно, и нет никаких гарантий, что туда не попадет нормальный хост (на котором, может быть, в какой-то момент поселился вирус, рассылающий спам, но теперь вирус уже вылечили). Вторая причина более банальна: предложенный выше механизм фильтрации гораздо эффективней любых DNSBL и при этом не полагается на непроверенные данные от третьих лиц.

## Посмотрим с другой стороны баррикад

Фильтровать спам научились, теперь же я постараюсь свести воедино всю информацию на тему того, что же надо делать, чтобы не попасть в спам.



### Для администраторов почтовых серверов:

- Всегда делайте MX записи ссылающимися на A-записи.
- A-запись для почтового сервера всегда должна иметь зеркальную PTR запись.
- Хост из HELO заголовка должен иметь A- или MX запись.
- Всегда создавайте SPF записи (да-да, это-то как раз не обязательно, но просто правило хорошего тона).
- Для всех писем, рассылаемых из обслуживаемого домена, ящик для обратного адреса должен существовать и принимать почту.



### Для тех, кто рассылает почту (из программ, с сайтов и т. д.):

■ Всегда посылайте почту только с существующим обратным адресом.

■ Никогда не посылайте почту с неподконтрольного вам домена, не проверив правила SPF для него. Например gmail.com в текущий момент позволяет посылать письма от его имени любому серверу, а вот yandex.ru и mail.ru сообщают через SPF о том, что посылка от их имени со сторонних серверов должна вызывать на себя пристальное внимание, что истолковывается умными спам-фильтрами как увеличение уровня оценки спама для данного письма.

■ Никогда не посылайте почту через неправильно настроенные SMTP серверы. Проверить сервер «на вшивость» по списку выше — дело 5 минут, вам поможет утилита dig или nslookup.

## Резюме

Конечно, предложенные настройки фильтруют не весь спам. Поэтому вполне возможно вам потребуется дополнительно использовать еще и контекстный фильтр, который будет анализировать содержание писем, например, spamassassin, хотя я ничего подобного и не использую. Однако при настройке почты все равно не забывайте про описанное в этой статье, потому что введенные параметры позволяют снизить нагрузку на сервер на порядки по отношению к использованию только контекстных фильтров, да и дополнительно обеспечивают отличную фильтрацию.



*Вы должны сами решить, какие ограничения вы готовы поставить на свой сервер, а какие нет. Многие скептически относятся ко многим представленным выше проверкам. Однако все они используются на реальных SMTP серверах в интернете, поэтому даже если вы включите вообще все, то будете далеко не одиноки. Поэтому если вы заметили сервер, который настроен неверно, не полнитесь отправить его администратору письмо на эту тему. Возможно вы сможете ему избежать гнева со стороны пользователей, чья корреспонденция не дошла (если его уже не выгнали взащей с работы к моменту написания вами письма). И никогда не забывайте, что ломанные хосты можно добавлять в вайт-лист, дабы принимать от них почту без проверок. ■*



# Удаленная переустановка Linux на примере Debian

## Автор

Алексей Смирнов  
Системный администратор,  
программист  
Компания SevStar  
г. Севастополь  
<http://soar.name>

Существует множество «фичей» упрощающих жизнь системному администратору. Различные системы удаленного управления позволяют установить ОС на девственно-чистый сервер и сегодня это совсем не проблема. Даже наличие второго винчестера превращает проблему переустановки операционки в банальность. Поэтому в этой статье рассмотрим самый суровый из случаев: пусть на сервере имеется только один полностью разбитый винчестер, а наша задача — удаленно переустановить ОС. Удивленно поднимем брови, умиляясь тому, как гибка наша любимая операционка, засучим рукава и приступим к делу.

В основе данного метода лежит идея о том, что мы можем использовать SWAP-раздел для установки временной операционной системы, а когда она встанет на ноги, то заняться основной. Главное и единственное, что нам нужно — это своп размером не менее 420 Мб (именно столько занимает lenny со всем необходимым).

В статье используются следующие допущения:

- у вас стоит дебиан, и ставить вы планируете дебиан,
- используется grub установленный в MBR.

Краткий план наших работ будет выглядеть так:

- превращаем swap-раздел в ext3;
- устанавливаем на него чистую ОС;

- перезагружаемся в нее;
- делаем нужные изменения на основном разделе;
- копируем чистую ОС из временного в основной раздел;
- загружаемся с основного раздела, включаем swap.

## Подготовка раздела

Первое, что мы сделаем, это убедимся, есть ли у нас этот своп-раздел вообще:

```
# free -m
```

```
...
```

```
Swap:          470          0          470
```

Как видим — есть, и размер (отображается в Мб) вполне удовлетворяет требованиям. Осталось выяснить, как у нас разбит диск:

```
# fdisk -l /dev/sda
```

```
...
```

| Device    | Boot | Start | End | Blocks   | Id | System               |
|-----------|------|-------|-----|----------|----|----------------------|
| /dev/sda1 | *    | 1     | 462 | 3710983+ | 83 | Linux                |
| /dev/sda2 |      | 463   | 522 | 481950   | 5  | Extended             |
| /dev/sda5 |      | 463   | 522 | 481918+  | 82 | Linux swap / Solaris |

Видно, что на sda1 текущая ОС, на sda5 своп. Запутаться сложно, но всякое бывает.

Отключаем своп:

```
# swapoff -a
```

Убедиться, что мы его выключили, можно выполнив все тот же free:

```
# free
```

```
...
```

```
Swap:          0          0          0
```

Обновим нашу таблицу разделов:

```
# fdisk /dev/sda
```

```
Command (m for help): t
```

```
Partition number (1-5): 5
```

```
Hex code (type L to list codes): 83
```

```
Changed system type of partition 5 to 83 (Linux)
```

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
WARNING: Re-reading the partition table failed with error 16:
Устройство или ресурс занят.
The kernel still uses the old table.
The new table will be used at the next reboot.▶
```

```
Syncing disks.
```

Нам заботливо сообщили, что ядро не увидит изменений до перезагрузки, но нам это пока и не нужно. Теперь самое время подготовить файловую систему на нашем старом новом разделе. Например, ext3:

```
# mke2fs -j /dev/sda5
```

```
Writing inode tables: done
```

```
Creating journal (4096 blocks): done
```

Примонтируем раздел куда-нибудь, и на этом его подготовка будет закончена.

```
# mkdir /mnt/temp
```

```
# mount /dev/sda5 /mnt/temp
```

```
# df -h
```

| Файловая система | Разм | Исп  | Дост | Исп% | смонтир. на |
|------------------|------|------|------|------|-------------|
| /dev/sda1        | 3,5G | 600M | 2,8G | 18%  | /           |

```
...
```

|           |      |     |      |    |           |
|-----------|------|-----|------|----|-----------|
| /dev/sda5 | 456M | 11M | 422M | 3% | /mnt/temp |
|-----------|------|-----|------|----|-----------|

## Установка «временной» ОС

Здесь нужно заметить, что временной она является только по своему местоположению. Чтобы дважды не прогонять все действия по сборке чистой системы в дальнейшем, мы просто скопируем ее на основной раздел.

Используем отличное средство для получения минимальной установки – debootstrap. Здесь и далее мы будем считать что проблем с интернетом на сервере нет (иначе какой же он после этого сервер?), поэтому выкачиваем все из репозитория.

```
# aptitude install debootstrap
```

```
...
```

```
Настраивается пакет debootstrap (1.0.10lenny1) ...
```

Данной утилите достаточно передать четыре параметра: желаемая архитектура, название релиза, директория установки и ссылка на полное зеркало. Архитектуру можно определить по выводу ``uname -a``, дистрибутив выбираем на свой вкус, директория в данном случае та, куда мы смонтировали наш раздел, а ссылку на зеркало можно взять здесь: <http://www.debian.org/mirror/list>.

Получается, что строка запуска выглядит примерно так:

```
# debootstrap --arch i386 lenny /mnt/temp
http://ftp.ru.debian.org/debian/
```

После нажатия **Enter** начнется процесс загрузки и установки пакетов, при достаточном среднем интернет-соединении (~10 Мбит) на это уходит порядка 5-10 минут. В конце вы увидите сообщение о том, что система успешно установлена:

```
I: Base system installed successfully.
```

Теперь начинается одна из самых ответственных процедур: нам нужно правильно сконфигурировать новую систему. Любому сисадмину, наверное, снились кошмары о том, как он теряет удаленный контроль над машиной, да и в конце концов все это мы затевали именно для того, чтобы не ехать в дата-центр. Поэтому отложим кружку кофе и сосредоточимся.

Сначала скопируем все важные настройки. Наверное, у каждого найдутся достаточно важные файлы, которые лежат не там где положено. У меня, к примеру, есть некий `/etc/rc.-routes` со всеми нестандартными маршрутами. Главное не забыть ничего. Приводить тут какой-либо список, мне кажется, совершенно бессмысленно, но у меня это выглядит примерно так:

```
# cp /etc/{resolv.conf,hosts,rc.local} /mnt/temp/etc
```

```
# cp /etc/network/interfaces /mnt/temp/etc/network
```

```
# cp /etc/your-stuff /mnt/temp/etc
```

В `fstab` напишем самое необходимое – `proc` и наш кор-

невой раздел:

```
# cat > /mnt/temp/etc/fstab << «#EOF»
> proc /proc proc defaults 0 0
> /dev/sda5 / ext3 errors=remount-ro 0 1
> #EOF
```

Теперь смонтируем `dev`-окружение, перейдем в чрут нашей временной системы и сразу примонтируем `proc`:

```
# mount --bind /dev /mnt/temp/dev
```

```
# chroot /mnt/temp /bin/bash
```

```
# mount -t proc proc /proc
```

Очевидно, нам понадобится менеджер пакетов, думаю рассказывать о его настройке отдельно не нужно.

```
# wget http://debian.soar.name/sources.list
-O /etc/apt/sources.list
```

```
# aptitude update
```

Настроим часовой пояс:

```
# dpkg-reconfigure tzdata
```

Также нам понадобятся следующие пакеты:

```
# aptitude install locales
```

```
# dpkg-reconfigure locales
```

```
# aptitude install console-data
```

```
# aptitude install ssh
```

```
# aptitude install sudo
```

Сразу же, чтобы не забыть, создадим пользователя и назначим ему пароль, иначе в `ssh` нас потом не пустят:

```
# adduser --ingroup users soar
```

```
# visudo
```

Теперь переустановим загрузчик. Сначала необходимо создать все файлы загрузчика на новом диске:

```
# aptitude install grub
```

```
# grub-install /dev/sda
```

После чего необходимо инициализировать МБР на загрузку с нашего нового раздела. Для этого все там же, в чруте, войдем в консоль граба и напишем следующее:

```
# grub
```

```
grub> root (hd0,<TAB>)
```

```
Possible partitions are:
```

```
Partition num: 0, Filesystem type is
ext2fs, partition type 0x83
```

```
Partition num: 4, Filesystem type is
ext2fs, partition type 0x83
```

Своеобразный автокомплит по табу подскажет нам, какие разделы есть в нашем распоряжении. Как видим все на единицу меньше чем в названии в системе. Инициализируем загрузку с нашего `sda5`:

```
grub> root (hd0,4)
```

```
Filesystem type is ext2fs, partition type 0x83
```

```
grub> setup (hd0)
```

```
...
```

```
Done.
```

```
grub> quit
```

Загрузчик установлен куда нужно. Теперь выберем и установим подходящее ядро:

```
# aptitude search linux-image
```

```
# aptitude install linux-image-2.6.26-2-686 ▶
```



В ходе установки нас спросят **Create a symbolic link to the current kernel image?**, на что мы ответим утвердительно. Так же сообщат, что мы устанавливаем ядро, требующее от загрузчика поддержку initrd, и уточнят, не передумали ли мы. Отвечаем **Нет** и установка заканчивается. Осталось обновить меню загрузчика:

```
# update-grub
Found kernel: /boot/vmlinuz-2.6.26-2-686
Updating /boot/grub/menu.lst ... done
```

Выходим из чрута, собираем нервы в кулак и отправляем сервер в первую перезагрузку:

```
# exit
# reboot
```

Если все было сделано правильно, то машина перезагрузится в нашу временную систему. Мы можем зайти в ssh как пользователь, которого мы только что добавили.

## Перенос системы на основной раздел

Форматируем и монтируем наш старый раздел:

```
# mke2fs -j /dev/sda1
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
# mkdir /mnt/temp
# mount /dev/sda1 /mnt/temp
```

К слову, на этом этапе можно провести и обслуживание диска: например переразбить основной раздел и проверить файловую систему.

Остается скопировать нашу чистую ОС на основной раздел:

```
# cp -a -x / /mnt/temp/
```

Обновляем fstab. На этот раз причешем его по всем правилам:

```
# cat > /mnt/temp/etc/fstab << «#EOF»
> # /etc/fstab: static file system information.
> #
> # <file system><mount point><type><options> <dump> <pass>
> proc /proc proc defaults 0 0
> /dev/sda1 / ext3 defaults,errors=remount-ro 0 0
> #EOF
```

В очередной раз нужно обновить граб, но теперь для загрузки уже с нашего основного раздела:

```
# mount --bind /dev /mnt/temp/dev
# chroot /mnt/temp/ /bin/bash
# grub-install /dev/sda
# grub
```

```
grub> root (hd0,0)
```

```
Filesystem type is ext2fs, partition type 0x83
```

```
grub> setup (hd0)
```

```
Checking if «/boot/grub/stage1» exists... yes
```

```
Checking if «/boot/grub/stage2» exists... yes
```

```
Checking if «/boot/grub/e2fs_stage1_5» exists... yes
```

```
Running «embed /boot/grub/e2fs_stage1_5 (hd0)»... 17 sectors are embedded.
```

```
succeeded
```

```
Running «install /boot/grub/stage1 (hd0) (hd0)1+17 p (hd0,0)/boot/grub/stage2 /boot/grub/menu.lst»... succeeded
```

```
Done.
```

```
grub> quit
```

```
# update-grub
```

```
Updating /boot/grub/menu.lst ... done
```

Тут стоит отметить, что у меня в этом месте, в отличие от первого случая, menu.lst обновляться отказался, и там по-прежнему оставались ссылки на sda5. Почему это происходит я так и не разобрался, поэтому вручную подредактировал этот файл:

```
# sed -i -e 's/sda5/sda1/g' /boot/grub/menu.lst
# sed -i -e 's/(hd0,4)/(hd0,0)/g' /boot/grub/menu.lst
```

Выходим из чрута и делаем вторую перезагрузку:

```
# exit
# reboot
```

После перезагрузки можно убедиться, что мы снова на родном основном разделе:

```
# df -h
Файловая система Разм Исп Дост Исп% смонтир. на
/dev/sda1 3,5G 436M 2,9G 13% /
```

Остается создать и включить своп:

```
# fdisk /dev/sda
```

```
Command (m for help): t
```

```
Partition number (1-5): 5
```

```
Hex code (type L to list codes): 82
```

```
Changed system type of partition 5 to 82 (Linux swap / Solaris)
```

```
...
```

```
The new table will be used at the next reboot.
```

```
Syncing disks.
```

```
# cat >> /etc/fstab << «#EOF»
```

```
> /dev/sda5 none swap sw 0 0
```

```
> #EOF
```

```
# mkswap /dev/sda5
```

```
# swapon -a
```

Убедимся что все нормально:

```
# free -m
```

```
Swap: 470 0 470
```

Ну и в конце, если вы редактировали /boot/grub/menu.lst вручную, то стоит все-таки запустить скрипт его обновления еще раз:

```
# update-grub
```

```
Updating /boot/grub/menu.lst ... done
```

## Подводя итоги

Данный способ, безусловно, не самый простой путь переустановки системы, однако во многих случаях он становится единственно возможным. Тем более вся процедура занимает около 25 минут, поэтому если ваш ДЦ не через дорогу, то в любом случае выходит совсем неплохая экономия времени. Получилось на удивление много букв – я пытался расписать подробно и понятно, но на самом деле операция простая и достаточно быстрая. ■

Автор

Алексей Ворошилов  
Студент ВКИ НГУ  
г. Новосибирск  
redsh.ru

# Создание полноценного сервера на базе недорогого домашнего роутера

Не секрет, что стоимость роутера значительно зависит от возможностей встроенного программного обеспечения. Так что покупая недорогое устройство многие пользователи не ждут от родной прошивки ничего хорошего и устанавливают альтернативную прошивку. Но, как правило, на этом все и заканчивается. Пользователь изучает встроенный веб-интерфейс и радуется новым возможностям поумневшего устройства. Однако прошивка роутера – это полноценная операционная система на базе Linux, позволяющая воплотить многие задумки, выходящие далеко за рамки привычной функциональности роутера.

В этой статье я хочу поделиться своим опытом по созданию сервера на основе роутера D-Link DIR-320. На роутере были подняты веб-сервер, FTP, настроена видеотрансляция. Хотя статья и предполагает использование DIR-320, для другого роутера процесс настройки будет отличаться незначительно.

## Аппаратное обеспечение

Я использовал роутер DIR-320. Данный девайс имеет MIPS32-микроконтроллер Broadcom BCM5354 с тактовой частотой 240 МГц, оперативную память объемом 32 МБ, энергонезависимую память объемом 4 МБ. Сетевые интерфейсы, 2.4 ГГц RF-модуль, USB-контроллер интегрированы в микроконтроллер. Еще в устройстве установлен 100-мегабитный свитч на 4 порта. Кстати, микроконтроллер достаточно сильно грелся, но после того, как на него был посажен на термопасту небольшой медный радиатор, температура снизилась до нормальных 40-50 градусов. Кроме этого, на плате не был установлен экран на высокочастотном модуле, хотя и была предусмотрена площадка для его пайки. Впрочем, эти небольшие недостатки конструкции компенсируются сравнительно низкой ценой

устройства, хорошим потенциалом для моддинга, поддержкой альтернативных прошивок, и легкостью восстановления стандартной прошивки после неудачных экспериментов. Кроме роутера был использован 4-портовый USB-хаб Manhattan, UVC-камера A4-Tech PK-750MJ на китайской микросхеме VC0332, USB-флешка Kingston на 2 ГБ, пара патч-кордов по 1,5 м. Питание осуществляется от стандартного блока питания на 2 Ампера из комплекта роутера. Запаса мощности вполне хватает для питания дополнительных устройств. При подключении более серьезных потребителей (например, HDD) желательно установить блок питания поприличней (рис. 1).

## Выбор прошивки

Первой прошивкой, на которую я обратил внимание, была DD-WRT. Эта прошивка порадовала качеством, стабильностью и обширным набором дополнительных пакетов Optware. DD-WRT поддерживает множество устройств, легко

устанавливается, имеет удобный веб-интерфейс, множество возможностей доступно сразу после заливки прошивки в роутер. Однако DD-WRT основана на старом ядре 2.4, что накладывает серьезные ограничения. Например, ядро 2.4 не поддерживает модуль uvcvideo, который требовался мне для подключения камеры. Погуглив, я узнал о прошивке OpenWrt. Последний релиз OpenWrt 10.03 под кодовым именем Backfire, вышедший 8 апреля 2010 г, поддерживает ядро 2.6.32. Для этой прошивки создан набор дополнительных пакетов X-Wrt, в котором можно найти очень много полезного портированного софта и админку Webif. Главная особенность OpenWrt – перезаписываемая файловая система, которая монтируется поверх стандартной файловой системы, что позволяет устанавливать пакеты в роутер и выполнять конфигурирование привычным редактированием файлов. Возможно монтирование как части энергонезависимой памяти роутера незанятой прошивкой (с использованием файловой системы ►

Рис. 1



JFFS2), так и внешнего носителя. Для установки пакетов используется менеджер орпк. OpenWrt вместе с дополнительными пакетами распространяется в исходниках.

## Сборка прошивки

Нам понадобится любая операционная система на основе Linux. Я использовал Arch Linux. Для получения исходников, нужно установить subversion:

```
$ sudo pacman -S subversion
```

Скачиваем Buildroot OpenWrt:

```
$ mkdir ~/openwrt && cd ~/openwrt
```

```
$ svn co svn://svn.openwrt.org/openwrt/branches/backfire
```

Скачиваем и устанавливаем дополнительные пакеты и патчи:

```
$ cd backfire
```

```
$ ./scripts/feeds update && ./scripts/feeds install -a
```

Проверяем наличие всех нужных для сборки пакетов и инициализируем конфигурацию прошивки:

```
$ make defconfig
```

Конфигурируем прошивку:

```
$ make menuconfig
```

В процессе конфигурирования нужно выбрать систему, пакеты, и настроить будущую прошивку. В принципе базовая конфигурация прошивки вполне работоспособна, нам нужно просто доконфигурировать прошивку по вкусу (рис. 2).

Выбираем систему:

```
Target System -> BCM947xx/953xx -- выбираем вариант для нашего BCM5354 с ядром 2.6
```

Выбирать пакеты можно как для установки в прошивку, так и просто для сборки. Пакеты, собранные без установки образуют наш мини-репозиторий, из которого их можно будет устанавливать уже после заливки прошивки в роутер. Памяти в DIR-320 всего 4 МБ, причем обязательно должно остаться немного свободного места для использования JFFS2. Так что будем устанавливать минимально необходимый набор пакетов, а остальное соберем для последующей установки на внешний носитель. Чтобы выбрать пакет для установки в прошивку, нужно нажать Y. Я добавил в прошивку пакеты, требуемые для монтирования флешки:

```
Kernel Modules -> USB Support -> core, usb2, ohci -- для поддержки контроллеров USB 2.0
```

```
Kernel Modules -> USB Support -> storage -- для поддержки USB-накопителей
```

```
Kernel Modules -> Filesystems -> ext3 -- выбираем нужные файловые системы
```

```
Base System -> block-mount -- для монтирования через fstab
```

```
Utilities -> disc -> block-extroot -- для монтирования поверх файловой системы
```

```
Utilities -> Editors -> nano -- удобный текстовый редактор для конфигурирования системы
```

Также, добавил веб-админку. Без админки не удастся установить пароль для входа по ssh:

```
Administration -> webif -> webif, webif-applications -- веб-админка
```

Затем, выбираем пакеты для сборки без установки. Не стоит выбирать много лишних пакетов, чтобы время сборки не увеличивалось до неразумных пределов. Чтобы выбрать пакет для сборки без установки, нужно нажать M. Многие пакеты можно установить и из основного репозитория, однако нужные модули ядра стоит собрать, т. к. в репозитории могут быть несовместимые версии. Я выбрал модули ядра, необходимые для подключения UVC-камеры – kmod-video-core и kmod-video-uvc.

Также можно настроить некоторые параметры прошивки, например сетевой адрес по умолчанию. Я установил адрес 192.168.0.1 (адрес роутера по умолчанию – 192.168.1.1):

```
Image Configuration -> LAN IP Address -> 192.168.0.1
```

При необходимости, можно сконфигурировать ядро. Я оставил все по умолчанию.

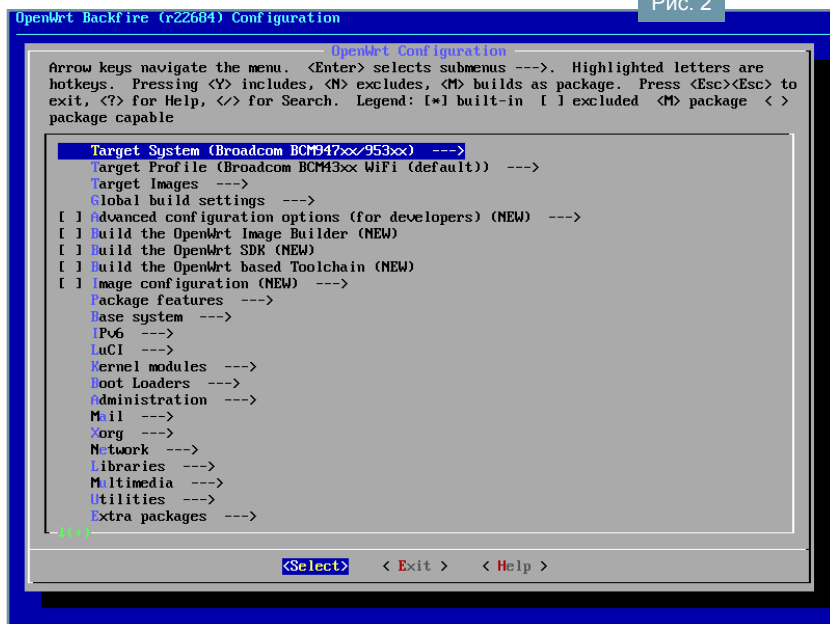
```
$ make kernel_menuconfig
```

Собираем прошивку:

```
$ make
```

В процессе сборки прошивки также будут скачиваться и собираться необходимые инструменты для кросс-компиляции, скачиваться исходники ядра, пакетов и т. д. Сборка займет от 30 минут до нескольких часов, в зависимости от производительности сети и компьютера. После завершения сборки, в папке bin будут находиться созданные образы прошивки в различном формате и собранные пакеты. Нам понадобится образ прошивки в формате squashfs – openwrt-brcm47xx-squashfs.trx.

Рис. 2



## Заливка прошивки

Для заливки прошивки понадобится TFTP-клиент (неважно, под Linux или Windows), я использовал GUI'шную утилитку под Windows, которую можно найти на сайте DD-WRT. Нужно подключить PC к LAN-порту роутера, выставить на сетевом интерфейсе PC статический IP 192.168.0.10 и маску подсети 255.255.255.0. В TFTP-клиенте выбрать файл прошивки и выставить IP сервера 192.168.0.1. Включить роутер и дождаться линка, затем отправить прошивку в роутер. Загрузчик ждет прошивку в течение пары секунд, так что возможно понадобится несколько попыток, также при необходимости стоит сразу разобраться с фаерволом, чтобы отправить прошивку без лишней задержки. После того, как прошивка



будет отправлена на роутер, нужно подождать пару минут пока загрузчик не завершит внутренний цикл прошивки и не включится диагностический светодиод.

Перед заливкой новой прошивки стоит запастись оригинальной прошивкой – если заливка пройдет неудачно, можно будет вернуть оригинальную прошивку через консоль восстановления. Для этого нужно включить роутер, удерживая **Reset**, затем зайти браузером на 192.168.0.1.

После завершения прошивки, настраиваем сетевой интерфейс компа на получение адреса с помощью DHCP, перезагружаем роутер. OpenWrt загружается несколько дольше оригинальной прошивки, причем диагностический светодиод светится во время загрузки и гаснет при завершении. Если все прошло успешно, заходим браузером на роутер. На этом этапе не обязательно настраивать роутер – после подключения внешнего накопителя, настройки будут храниться на нем и придется настраивать заново. Достаточно лишь установить пароль.

## Подготовка накопителя

Накопитель нужно разбить на разделы fdisk'ом и отформатировать. Я создал 3 раздела – под систему, под домашние директории и под своп. Не забудьте заменить /dev/sdb на имя съемного накопителя в своей системе:

```
Disk /dev/sdb: 2004 MB, 2004877312 bytes
62 heads, 62 sectors/track, 1018 cylinders
Units = cylinders of 3844 * 512 = 1968128 bytes
Disk identifier: 0xa8afelb3
```

```
Device Boot Start End Blocks Id System
/dev/sdb1 1 255 488281+ 83 Linux
/dev/sdb2 255 992 1416015+ 83 Linux
/dev/sdb3 992 1019 52567 82 Linux swap / Solaris
```

Форматируем:

```
$ mkfs.ext3 /dev/sdb1
$ mkfs.ext3 /dev/sdb2
$ mkswap /dev/sdb3
```

## Настройка прошивки для загрузки с внешнего накопителя

Заходим на роутер SSH-клиентом используя логин root и пароль, установленный через веб-админку. Редактируем файл /etc/config/fstab.

```
$ nano /etc/config/fstab
```

Редактируем одну секцию «mount» для монтирования раздела накопителя поверх основной файловой системы. Для монтирования других разделов, будет использоваться файл fstab, хранящийся на внешнем накопителе:

```
config mount
```

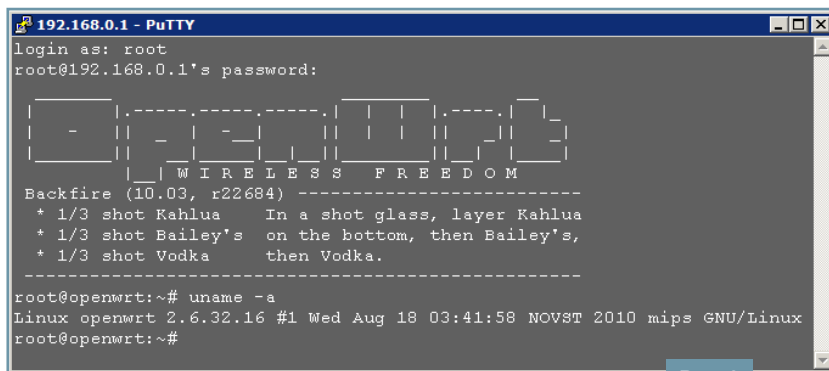


Рис. 3

```
option target /
option fstype ext3
option enabled_fsck 0
option device /dev/sda1
option options rw, sync
option is_rootfs 1
option enabled 1
```

Перезагружаем роутер, заходим в веб-админку. Если все прошло успешно, админка должна снова попросить установить пароль – устройство успешно смонтировалось и настройки теперь хранятся на нем.

## Настройка прошивки

Заходим на роутер SSH-клиентом (рис. 3), создаем точку монтирования для домашних директорий:

```
$ mkdir /home
```

При желании можно установить собранные вместе с прошивкой пакеты: расшариваем их (папку bin Buildroot'a) FTP-сервером, затем прописываем наш мини-репозиторий в файле /etc/opkg.conf, добавив строку:

```
src/gz mypackages ftp://(адрес сервера)/brmc47xx/packages
```

Обновляем список пакетов:

```
$ opkg update
```

Если при обновлении будут получены списки пакетов из основного репозитория, их можно удалить:

```
$ cd /var/opkg-lists
$ rm X-Wrt packages
```

Устанавливаем нужные пакеты:

```
$ opkg install kmod-input-core kmod-video-core kmod-video-uvc
```

Основное конфигурирование удобно выполнять через веб-админку. Устанавливаем имя хоста, часовой пояс, добавляем точки монтирования, настраиваем сетевые интерфейсы и т. д. Я смонтировал раздел для домашних директорий с опциями `async` и `noatime`, чтобы продлить жизнь флешке. Перезагружаем роутер и проверяем результат:

```
$ df -h
```

```
Filesystem Size Used Available Use% Mounted on
/dev/root 2.0M 2.0M 0 100% /rom
tmpfs 14.5M 72.0K 14.5M 0% /tmp
tmpfs 512.0K 0 512.0K 0% /dev
/dev/sda1 461.8M 19.8M 418.1M 5% /overlay
mini_fo:/overlay 2.0M 2.0M 0 100% /
```

```
/dev/sda2 1.3G 558.4M 733.6M 43% /home
```

## Устанавливаем FTP-сервер

Создадим группу админов сервера:

```
$ echo web:x:101: >> /etc/group
```

Создадим пользователя ftp для анонимного входа, добавим его в группу nogroup (65534), и пользователя ftpadmin для администрирования файлов, доступных по FTP.

```
$ echo ftpadmin*:102:101:ftpadmin:/home/srv/ftp:/bin/ash >> /etc/passwd
```

```
$ echo ftp*:103:65534:ftp:/home/srv/ftp:/bin/ash >> /etc/passwd
```

```
$ mkdir -p /home/srv/ftp
```

```
$ chgrp -R web /home/srv
```

```
$ chown ftpadmin /home/srv/ftp
```

Зададим пароль для пользователя ftpadmin

```
$ passwd ftpadmin
```

Установим pure-ftpd – быстрый и легкий FTP-сервер.

```
$ opkg update
```

```
$ opkg install pure-ftpd
```

Настройки сервера находятся в файле /etc/config/pure-ftpd. Разрешим анонимный вход и выберем отдельный лог, отредактировав следующие строки:

```
option noanonymous '0'
```

```
option anonymouscantupload '1'
```

```
option syslogfacility 'none'
```

```
option altlog 'stats:/var/log/pureftpd.log'
```

```
option enabled '1'
```

Также удалим опцию peruserlimits, чтобы убрать ограничения на количество входов пользователей.

Запустим сервер:

```
$ /etc/init.d/pure-ftpd start
```

Зайдем в веб-админку на вкладку **Services** и отметим pure-ftpd для автоматического запуска. Чтобы сервер был доступен из интернета, нужно зайти на вкладку **Firewall** и открыть 21 порт.

Рис. 4



## Устанавливаем веб-сервер

Создадим домашнюю директорию для файлов и пользователя, который будет владеть этой директорией:

```
$ echo wwwadmin*:101:101:wwwadmin:/home/srv/www:/bin/ash >> /etc/passwd
```

```
$ mkdir /home/srv/www
```

```
$ chgrp web /home/srv/www
```

```
$ chown wwwadmin /home/srv/www
```

Зададим пароль для пользователя wwwadmin:

```
$ passwd wwwadmin
```

Установим веб-сервер lighttpd с нужными плагинами:

```
$ opkg update
```

```
$ opkg install lighttpd lighttpd-mod-cgi lighttpd-mod-access lighttpd-mod-redirect
```

Установим PHP:

```
$ opkg install libsqlite3 php5-cgi php5-mod-pdo php5-mod-sqlite php5-mod-gd php5-mod-session
```

Сконфигурируем lighttpd. Настройки хранятся в файле /etc-/lighttpd/lighttpd.conf.

Подключим нужные нам плагины:

```
server.modules = (
```

```
«mod_cgi»,
```

```
«mod_access»,
```

```
«mod_redirect»
```

```
)
```

Установим директорию для документов:

```
server.document-root = «/home/srv/www»
```

Зададим имена документов по-умолчанию:

```
index-file.names = ( «index.htm», «index.php» )
```

Укажем CGI-обработчик для PHP:

```
cgi.assign = ( «.php» => «/usr/bin/php-cgi» )
```

Желательно указать UID и GID для сервера, чтобы он не работал с правами суперпользователя:

```
server.username = «wwwadmin»
```

```
server.groupname = «web»
```

Настроим PHP, отредактировав файл /etc/php.ini. Подключим нужные модули, раскомментировав соответствующие строки:

```
extension=pdo.so
```

```
extension=gd.so
```

```
extension=session.so
```

```
extension=sqlite.so
```

Изменим параметр doc\_root на пустую строку, иначе интерпретатор будет искать скрипты не в том месте:

```
doc_root =
```

Чтобы сервер заработал, нужно сначала перевесить веб-админку роу- ▶

тера с 80 порта на какой-нибудь другой порт. Запустим сервер:

```
$ /etc/init.d/lighttpd start
```

Создадим в /home/srv/www файл test.php с таким содержанием:

```
<?php phpinfo(); ?>
```

Откроем 192.168.0.1/test.php. Если все прошло успешно, увидим таблицу с настройками PHP.

Перейдем на вкладку **Services** веб-админки и разрешим автоматический запуск lighttpd. На вкладке **Firewall** откроем 80 порт, чтобы сервер был виден из интернета.

## Настраиваем видеотрансляцию

Для начала, надо установить нужные для камеры модули и убедиться, что камера успешно подхвачена системой:

```
$ ls /dev/video*
```

```
/dev/video0
```

Для видеотрансляции будем использовать пакет MJPEG-Streamer. Это довольно тяжелый пакет, но его можно немного «твикнуть».

Вернемся к Buildroot'у. Получим исходники пакета:

```
$ cd ~/openwrt/backfire
```

```
$ make package/mjpg-streamer/clean
```

```
$ make package/mjpg-streamer/prepare
```

Поправим код для сжатия в JPEG. Путь может немного отличаться для разных версий пакета и библиотек:

```
$ nano build_dir/target-mipsel_uClibc-0.9.30.1/mjpg-streamer-r94/plugins/input_uvc/jpeg_utils.c
```

Найдем в коде строку инициализации параметров сжатия:

```
jpeg_set_defaults (&cinfo);
```

После этой строки можно переопределить некоторые параметры сжатия JPEG. Выберем быстрый целочисленный метод для DCT:

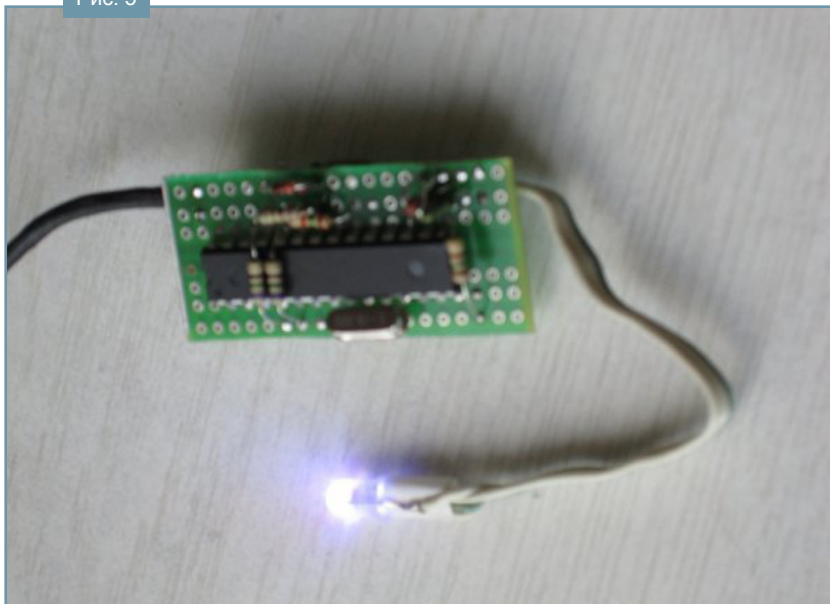
```
cinfo.dct_method = JDCT_IFAST;
```

Сохраним изменения, скомпилируем пакет, обновим индекс пакетов:

```
$ make package/mjpg-streamer/compile V=99
```

```
$ make package/index
```

Рис. 5



Подключимся к роутеру по SSH. Включим наш репозиторий в /etc/opkg.conf. Установим MJPEG-Streamer:

```
$ opkg update
```

```
$ opkg install mjpg-streamer
```

Настройки пакета находятся в файле /etc/config/mjpg-streamer. Отредактируем нужные строки:

```
config mjpg-streamer core
```

```
option device «/dev/video0»
```

```
option resolution «640x480»
```

```
option fps «1»
```

```
option port «310»
```

```
option enabled «true»
```

Если камера отдает поток в формате MJPG, на этом настройка закончена. Моя камера отдает несжатый поток в формате YUV, для нее нужно включить JPEG-сжатие. Отредактируем файл /etc/init.d/mjpg-streamer. Найдем командную строку запуска демона:

```
$PROG -- --input «input_uvc.so --device $device --fps $fps --resolution $resolution» --output «output_http.so --port $port» &
```

Добавим параметр --yuv для плагина input\_uvc:

```
$PROG -- --input «input_uvc.so --device $device --fps $fps --resolution $resolution --yuv» --output «output_http.so --port $port» &
```

Стартуем демон:

```
$ /etc/init.d/mjpg-streamer start
```

Если все прошло нормально, должен включиться светодиод на камере. Для проверки работы сервера зайдём на 192.168.0.1:310/?action=stream. Не все браузеры нормально поддерживают MJPG-поток. Статическая картинка доступна по адресу 192.168.0.1:310/?action=snapshot. Для отображения видео на веб-странице можно использовать встроенную поддержку браузера (имеется в Firefox, Chrome, частично в Opera), Javascript или Java-апплет (рис. 4).

## Индикация посетителей

Мне хотелось сделать какую-нибудь индикацию приходящих посетителей. Для этого я взял яркий RGB светодиод, подключил его к USB через самодельный переходник на микроконтроллере (рис. 5). Программу для управления написал и отладил в Visual C, затем портировал и собрал под роутер с помощью Buildroot'a OpenWrt. Приходящие посетители, активность веб-сервера и FTP и другие события обозначаются вспышками разных цветов.

## P. S. На всякий случай

Помните, что вскрытие и перепрошивка роутера лишают гарантии на девайс. Возможны и другие проблемы. Впрочем новые роутеры (например, DIR-320) сложно убить перепрошивкой — загрузчик зашит в энергонезависимую память микроконтроллера (в отличие, например, от RedBoot, использовавшегося в более ранних моделях), так что в любой момент можно сделать откат на оригинальную прошивку. Но в любом случае, вы «ковыряете» роутер на свой страх и риск. ■





# Каталог Хостинг-компаний

## Superhost.com.ua

Организация: SARBASH Lab.  
Страна: Украина  
Город: Донецк  
Дата-центр: McLean data center, DCA2  
Сайт: <http://superhost.com.ua>  
Телефон: 380622521461  
E-mail: [info@superhost.com.ua](mailto:info@superhost.com.ua)

## Инстант Хостинг

Организация: Inst-Host.ru  
Страна: Germany  
Город: Nuremberg  
Дата-центр: HETZNER Online  
Сайт: <http://inst-host.ru/>  
E-mail: [reg@inst-host.ru](mailto:reg@inst-host.ru)

## xTremeHost.ru

Организация: xTremeHost.ru  
Страна: Казахстан  
Город: Кокшетау  
Дата-центр: США  
Сайт: <http://xtremehost.ru>  
Телефон: +77770397355  
E-mail: [support@xtremehost.ru](mailto:support@xtremehost.ru)

## КОМТЕТ

Организация: ООО «КОМТЕТ»  
Страна: Россия  
Город: Москва, Пенза  
Дата-центр: Москва, М10  
Сайт: <http://komtet.ru>  
Телефон: 8-800-200-2511  
E-mail: [support@komtet.ru](mailto:support@komtet.ru)

## Мирекс

Организация: Мирекс  
Страна: Россия  
Город: Красноярск, Москва  
Дата-центр: Квиклайн, М9, Веб Дата-центр -  
Дата Центры в Москве.  
Сайт: <http://mirex.su>  
Телефон: 8 (391) 215-29-35  
E-mail: [hosting@mirex.su](mailto:hosting@mirex.su)

## Партизанск Телеком

Организация: ИП Нецадим А.С.  
Страна: Россия  
Город: Партизанск  
Дата-центр: Burst, Netdirect, SuperB  
Сайт: [www.p-telecom.ru](http://www.p-telecom.ru)  
Телефон: +7 42363 69663  
E-mail: [company@p-telecom.ru](mailto:company@p-telecom.ru)

## Хостинг Хутор

Организация: HostingHutor.com  
Страна: Украина  
Город: Одесса  
Дата-центр: Hetzner, Utel, Wnet  
Сайт: <http://hostinghutor.com>  
Телефон: +3 048 7017698  
E-mail: [info\[at\]hostinghutor.com](mailto:info[at]hostinghutor.com)

## HOST-FOOD

Организация: ИП Седов А.Н.  
Страна: Россия  
Город: Саратов  
Дата-центр: «Инфобокс» в СПб и  
«Datacheap» в Мск  
Сайт: <http://www.host-food.ru>  
Телефон: +7 927-279-3758  
E-mail: [support@host-food.ru](mailto:support@host-food.ru)

## Казахстанский Хостинг

Организация: Уральск Web  
Страна: Казахстан  
Город: Уральск  
Дата-центр: АО «Казахтелеком», г.Уральск  
Сайт: [www.solidhost.kz](http://www.solidhost.kz)  
E-mail: [support@solidhost.kz](mailto:support@solidhost.kz)

## InstantHost

Организация: Instant  
Страна: Россия  
Город: Братск  
Дата-центр: Майами  
Сайт: <http://instanhost.biz>  
Телефон: +79501171041  
E-mail: [iskander019@ya.ru](mailto:iskander019@ya.ru)

## Iphosters

Организация: Iphosters.com  
Страна: USA  
Город: New York  
Дата-центр: New York  
Сайт: <http://iphosters.com>  
Телефон: +1-877-745-6763  
E-mail: [support@iphosters.com](mailto:support@iphosters.com)

## KILOBYTE

Организация: KILOBYTE.COM.UA  
Страна: Украина  
Город: Киев  
Дата-центр: собственная площадка  
Сайт: <http://kilobyte.com.ua>  
Телефон: +380443600206  
E-mail: [sales\[at\]kilobyte.com.ua](mailto:sales[at]kilobyte.com.ua)

## StoreHost.ru

Организация: СторХост Лтд  
Страна: Россия  
Город: Москва  
Дата-центр: «Оверсан-Меркурий»  
Сайт: <http://storehost.ru>  
Телефон: +7 495 9164790  
E-mail: [support@storehost.ru](mailto:support@storehost.ru)

## SVAI

Организация: ФОП Полудень В.М.  
Страна: Украина  
Город: Черкасы  
Дата-центр: Mhost, Megastyle, Ukrtelecom  
Сайт: [svai.net](http://svai.net)  
Телефон: +38 0472 569026  
E-mail: [billing@svai.net](mailto:billing@svai.net)

## Amberhost.net

Организация: Amberhost.net  
Страна: Украина  
Город: Киев  
Дата-центр: Мхост  
Сайт: <http://amberhost.net>  
E-mail: [support@amberhost.net](mailto:support@amberhost.net)

## KirHost

Организация: СПД Кунев Д.В.  
Страна: Украина  
Город: Кировоград  
Дата-центр: Воля, SteepHost  
Сайт: [www.kirhost.com](http://www.kirhost.com)  
E-mail: [roman@kirhost.com](mailto:roman@kirhost.com)

## Luckhost.net

Организация: ТОО «КУЛЬСЫН»  
Страна: Казахстан  
Город: Тараз  
Дата-центр: Hetzner  
Сайт: <http://luckhost.net/>  
E-mail: [info@luckhost.net](mailto:info@luckhost.net)

## ProGoldHost.Net

Организация: ProGoldHost.Net  
Страна: Россия  
Город: Ставрополь  
Дата-центр: США - DataJersey / Германия -  
Hetzner  
Сайт: <http://progoldhost.net>  
E-mail: [info@progoldhost.net](mailto:info@progoldhost.net)

## ISPserver

Организация: ISPserver  
Страна: Россия  
Город: Москва, Иркутск  
Дата-центр: Веб-дата центр (WebDC.ru)  
Сайт: [www.ispserver.com](http://www.ispserver.com)  
E-mail: [sales@ispserver.com](mailto:sales@ispserver.com)

## Компания iHead

Организация: Компания iHead  
Страна: Россия  
Город: Киров  
Дата-центр: Multinex, Agava  
Сайт: [www.ihead.ru](http://www.ihead.ru)  
E-mail: [info@ihead.ru](mailto:info@ihead.ru)

## FirstVDS

Организация: ЗАО «Первый»  
Страна: Россия  
Город: Москва, Иркутск  
Дата-центр: Веб-дата центр (WebDC.ru)  
Сайт: [www.firstvds.ru](http://www.firstvds.ru)  
E-mail: [sales@firstvds.ru](mailto:sales@firstvds.ru)

Добавить хостинг-компанию в каталог бесплатно

ДОБАВИТЬ

Публикуемые в журнале компании взяты из каталога сайта издания. Данные о компаниях публикуются по мере их добавления в каталог. Добавить в каталог данные о компании может любая организация. Добавление данных в каталог сайта осуществляется бесплатно на добровольной основе самой организацией. Редакция журнала Веб-Аналитик.ИНФО не занимается добавлением данных в каталог сайта и не несет ответственности за предоставляемую информацию от организаций.



## CMS

**Интервью с  
Тимофеем Горшковым  
[ Генеральный директор  
компании Инсейлс ]**

**UMI.CMS: сайт  
как набор XML-сервисов**

**Учимся работать с  
компонентами и  
информационными блоками  
CMS 1С-Битрикс**



# Интервью с Тимофеем Горшковым

**[ Генеральный директор компании Инсейлс ]**

Беседовал: Станислав Горнаков

Облачные технологии или облачные решения (Cloud Computing) все больше и больше набирают популярность в сети Интернет. Ныне известная услуга SaaS (Software as a service – программное обеспечение как услуга) постепенно выходит на новый уровень своего развития. Множество компаний пытаются внедрять различные услуги в интернете на базе именно SaaS сервисов. Одной из таких компаний является компания InSales, специализирующаяся на предоставлении площадки для электронной торговли. Программная платформа InSales – это система управления сайтом для интернет-магазина, предоставляемая по модели SaaS. Сейчас этот вид услуг интересен многим, поэтому мы решили пообщаться с Тимофеем Горшковым, который является генеральным директором компании InSales.

**Станислав Горнаков:** *Добрый день, Тимофей. Приветствую вас на страницах журнала Веб-Аналитик.ИНФО. Тимофей, начнем с краткой характеристики вашего сервиса. Возможно, многие еще не в курсе или не понимают, что такое SaaS или «интернет-магазин в аренду», поэтому расскажите, пожалуйста, об истории создания компании InSales, чем компания занимается на данный момент, и какие услуги предлагает пользователям.*

**Тимофей Горшков:** Добрый день, Станислав. Компания InSales была организована 2,5 года назад группой единомышленников. Нашу команду объединяло обучение в одном ВУЗе – МФТИ, а успешно начать новый бизнес помог разносторонний бизнес-опыт, полученный членами команды в разных проектах: свой интернет-магазин спортивных товаров, дистрибуция цифровой тех-

ники, поисковое продвижение, управление командой разработки, продажа смс-сервисов. В 2008 году приближение кризиса висело в воздухе, и было понятно, что большое количество компаний в скором времени начнут поиск альтернативных каналов продаж. А наибольший рост можно ожидать именно в сфере интернет-торговли. Сейчас, обращаясь к результатам прошедших двух лет, видно, что именно так все и произошло. Даже в кризисный год оборот розничной торговли через интернет продолжал бурно расти. В интернет-продажи пришло много новых людей, много новых бизнесов.

В 2008 году мы подробно прорабатывали бизнес-модель одной из самых важных задач, которые мы ставили перед собой – это создание удобного бизнес-инструмента для ведения продаж в интернете обычными торговыми компаниями. Не программистами или интернет-гуру, а именно малым и средним бизнесом. Поэтому созданный нами инструмент – платформа InSales – разрабатывался как SaaS-сервис. Для офлайн бизнеса важно найти качественного исполнителя (компанию-аутсорсера), а не нанимать свою бригаду программистов и системных администраторов.

**С. Г.:** *Поясните для наших читателей что такое «аренда интернет-магазина» или SaaS-платформа для интернет-магазина? Как это связано с аутсорсингом и в чем отличие от покупки коробочной CMS?*

**Т. Г.:** SaaS расшифровывается как Software as a Service, т. е. программное обеспечение как услуга. За этими словами скрывается вполне простая вещь. Покупая SaaS, вы регулярно платите за использование программного обе-

спечения, которое поддерживается и развивается провайдером. Хорошим примером для понимания может быть интернет-банкинг. Вы платите регулярно какую-то сумму за то, чтобы использовать удобный онлайн-инструмент. Вы не думаете ни о каком администрировании, развитии и обновлении этого программного обеспечения – об этом думает банк.

В случае покупки коробочной CMS, вы платите разово за лицензию, а далее ежегодно покупаете обновления и новые версии. Также ежемесячно платите за хостинг, где будет установлена эта CMS, и за квалифицированное администрирование. За администрирование можно платить ежемесячно (зарплата системного администратора или плата за поддержку веб-студией), либо разово при возникших проблемах или необходимости установки обновлений.

В обоих случаях есть регулярные платежи, но в случае SaaS-сервиса они все сложены в одну понятную ежемесячную сумму. А в случае коробочной CMS, осознаются владельцем магазина по мере столкновения с каждой из них, и многие даже не делают оценку сколько реально ежемесячно тратится на поддержку решения для сайта интернет-магазина.

Я стараюсь избегать термина «аренда», потому что он не всегда правильно характеризует SaaS-платформу. На рынке представлены решения по аренде интернет-магазина, когда веб-студия принимает ежемесячную плату за обслуживание отдельно установленных копий интернет-магазинов своих клиентов. Хочу подчеркнуть, что это аренда, но не SaaS.

**С. Г.:** *Вроде эти термины обычно используются как синонимы. В чем отличие?* ▶



**Т. Г.:** Ключевым отличием SaaS-платформы на примере InSales является то, что это большая единая платформа для множества интернет-магазинов (как торговый центр с общей инфраструктурой для всех магазинов). Чем это выгодно для владельца интернет-магазина?

Во-первых, магазин хорошо масштабируется. Если в один день в него придут десятки и сотни тысяч посетителей, то для SaaS-платформы, на которой работают сотни магазинов это почти незаметно в общем объеме. А если это отдельно установленная веб-студией копия, то сайт «ляжет» или будет работать очень медленно, и все рекламные деньги будут потрачены зря.

Во-вторых, надежность работы. В SaaS-платформу заложены механизмы распределенного хранения данных, пресловутые «облачные» технологии. В результате выход из строя одного жесткого диска или одного сервера никак не скажется на работоспособности интернет-магазина. Что будет в случае отдельно стоящего арендуемого или собственного интернет-магазина, думаю, не нужно пояснять. И хотя сейчас активно рекламируются «облачные хостинги», рекомендую подробно расспрашивать, что подразумевают под этими словами.

**С. Г.:** То есть, подводя общий итог выше сказанного, у InSales купить программное обеспечение нельзя, можно только пользоваться на определенных условиях?

**Т. Г.:** Хочу подчеркнуть, что покупка программного обеспечения всегда совершается по лицензионному договору, в котором оговариваются условия использования приобретаемого ПО. Вы не сможете его распространять, перепродавать. Вы сможете его только использовать и иногда модифицировать. Программное обеспечение InSales тоже можно использовать и можно дополнять разрабатываемыми модулями-приложениями. Да, скорее всего в случае использования SaaS будет больше ограничений, но не надо забывать, что они компенсируются своими плюсами.

Еще важно разъяснить моменты по поводу владения сайтом интернет-магазина. Используя SaaS-платформу для интернет-магазина, предприниматель владеет дизайном и разработанной для его магазина логикой работы витрины. Эта часть отчуждаема и переносима на другие платформы. Также предприниматель владеет и всеми внешними в систему данными. Товарный каталог с картинками выгружаем, клиентская база и база заказов тоже. То есть на самом деле, используя SaaS-платформу, предприниматель владеет как визуальной частью сайта магазина, так и всеми коммерческими данными. Арендует он только платформу, остальным — владеет.

**С. Г.:** Большинство людей, с кото-

*рыми я общался или общаюсь, выражают опасения по поводу того, что сторонняя компания, предоставляющая SaaS сервис, может взять и закрыться, а мой бизнес полетит в тартары. То есть никто у нас в стране как-то не хочет особо доверять свой бизнес SaaS сервисам. Ведь, кажется куда проще и легче самому взять и поставить специализированную CMS на хостинг и продавать любой товар в Интернете, не так ли?*

**Т. Г.:** Опасения связаны с непониманием. Все неизвестное всегда пугает. По мере того, как люди будут лучше понимать, какие риски действительно у них есть в обоих случаях, доверие к SaaS-сервисам будет расти.

По поводу страха, что бизнес полетит в тартары. Надо разделять бизнес и отдельные используемые инструменты. Если арендодатель, где я снимаю офис повысит стоимость аренды и мне придется переехать, мой бизнес полетит в тартары? А если у хостинга, на котором у установил купленную CMS изымут сервера (как это было в недавнем прошлом у одного из участников рынка), мой бизнес тоже закроется? А если у меня заболит менеджер, принимающий заказы по телефону, тоже бизнес закрывать?

Предприниматель должен оценивать риски и при необходимости предпринимать шаги по их минимизации. Например, делать резервную копию дизайна шаблона интернет-магазина после существенных изменений. Иметь выгруженную на локальный компьютер базу товаров. Регулярно выгружать заказы в excel, а в идеале в учетную систему. Чем рискует в таком случае владелец интернет-магазина, используя SaaS? Максимум несколькими днями простоя сайта магазина, пока осуществляется загрузка этих сохраненных данных на другую платформу или CMS. Риски сравнимы с использованием корпоративной CMS и хостинга. Да и на время проблем на домене магазина можно

разместить заглушку с извинениями, контактами магазина и выложенным прайс-листом.

А по поводу того, что поставить CMS на хостинг проще, чем использовать SaaS, это вы точно погорячились. Имея базу товаров в excel, на платформе InSales можно запустить сайт интернет-магазина в течение получаса. Мы это демонстрируем на всех вебинарах. Куда уж проще?

**С. Г.:** Если не секрет, какое количество магазинов сейчас работает на платформе InSales?

**Т. Г.:** Более 300 компаний и предпринимателей ведут торговлю либо активно подготавливают магазин к продажам.

**С. Г.:** У вас имеется несколько разных тарифов, охарактеризуйте каждый из тарифов, в чем их отличия и с какого тарифа лучше начинать свою деятельность скажем начинающим бизнесменам.

**Т. Г.:** Если говорить об InSales как о платформе, то она предоставляется по 5 основным тарифным планам, различающимся количеством товарных позиций в интернет-магазине и объемом функциональности. Начиная с самого дешевого тарифа, мы предоставляем весь комплекс инструментов по организации витрины интернет-магазина. Более дорогие тарифы отличаются интегрированными инструментами повышения лояльности (личный кабинет, накопительные скидки) и синхронизацией с учетными системами.

На всех основных тарифах можно использовать свое доменное имя для магазина, создавать произвольный дизайн, получать техническую поддержку по email и по телефону. Для совсем небольших интернет-продавцов (например, для продажи товаров, сделанных своими руками, в блогах и социальных сетях) у нас есть тариф «Социальный» ▶

Продукты

Услуги

Клиенты

Поддержка

Партнёрам

О компании

Платформа интернет-магазина для Вашего бизнеса

InSales — профессиональное решение для создания магазина. Удобное и лёгкое в обслуживании.

✓ Мощная функциональность

✓ Регулярные обновления

✓ Хостинг и поддержка

Создать пробный магазин

14 дней бесплатно

Начинающему онлайн-бизнесу

готовый интернет-магазин

Работающему офлайн-бизнесу

комплексное решение

Готовые дизайн-шаблоны

и другие услуги магазинам

Почему лучше создавать интернет-магазин на системе InSales?

+1

Всё техническое включено

В тарифные планы включена стоимость программного обеспечения, регулярные обновления, надёжный хостинг и техническая поддержка вашего магазина.

+2

Сохранность ваших данных

Все ваши данные принадлежат вам. Если вы решите сменить систему, то сможете выгрузить из InSales все товары, покупателей и даже дизайн в виде zip-архива.

+3

Профессиональная забота

+4

Скорость и надёжность

Получить ответ на вопрос:

Выберите суть вопроса \*

Уточните вопрос

Введите ваше имя \*

Введите ваш e-mail \*

Введите ваш телефон

Отправить запрос

Веб-Аналитик.ИНФО сентябрь 2010

26

без абонентской платы, с ограниченной функциональностью и без возможности использовать свой домен.

**С. Г.:** Кстати, клиент вашей системы может иметь свой домен второго уровня, то есть домен такого вида: [www.моя\\_компания.ru?](http://www.моя_компания.ru?)

**Т. Г.:** Разумеется, клиент может иметь свой домен второго уровня. Название домена [моя\\_компания.myinsales.ru](http://моя_компания.myinsales.ru) используется как идентификатор аккаунта магазина в нашей системе, плюс временно используется пока не настроен свой домен. После прикрепления к магазину домена вида [www.моя\\_компания.ru](http://www.моя_компания.ru) и владелец, и покупатели работают с сайтом интернет-магазина на его домене второго уровня.

**С. Г.:** Интеграция с какими платежными системами реализована в InSales?

**Т. Г.:** На данный момент в InSales реализована интеграция с терминалами оплаты Qiwi, платежными системами Webmoney и PayPal, Robox, оплата по пластиковым картам может приниматься через системы LiqPay, Platron, PayOnline, Portmone.com, ActivePay, Assist. Ну и мы, конечно же, стараемся подключать все популярные и актуальные платежные системы, чтобы у клиентов был хороший выбор по условиям.

Можно сказать, что у нас один из самых широких выборов подключаемых способов оплаты. Более того, подключение этих систем доступно без дополнительной платы, в отличие от коробочных CMS, в которых отдельные модули под платежные системы бывают платными.

**С. Г.:** У вас есть и тестовый период на 14 дней для ознакомления с предоставляемыми услугами. Какие



функции магазина доступны пользователю в этот тестовый период?

**Т. Г.:** На весь тестовый период пользователь получает полную функциональность интернет-магазина в соответствии с выбранным тарифом. Допустим, вы создали тестовый магазин на тарифе «Стандартный», все две недели вы пользуетесь всем функционалом, доступным на данном тарифе. То же самое происходит, если вы создали магазин на тарифе «Бизнес+». То есть у нас нет ограничений в этом плане — независимо от того, какова абонентская плата за тариф, две недели вы можете пользоваться всеми функциями бесплатно.

**С. Г.:** Также у вас на сайте в качестве одного из продуктов я видел «Индивидуальные решения», что это за услуга?

**Т. Г.:** Это комплексное отраслевое решение под бизнес-задачу клиента. Если взять тематику одежды, автозапчастей или детских товаров, то в каждой из них при решении требуется не просто сайт интернет-магазина. В автозапчастях нужна специфическая функциональность подбора по марке машины. В одежде и детских товарах принципиальны email-рассылки и их интеграция с сайтом, и так далее. Обслуживая уже длительное время магазины по этим тематикам, мы очень хорошо представляем бизнес-потребности и их решения. Причем InSales в таком случае выступает в качестве интегратора, собирающего решение из «кубиков» под конкретный кейс клиента, исходя из бюджетов и стратегии продвижения магазина.

Еще одно интересное решение у нас есть в партнерстве с Лабораторией Касперского. Платформа InSales позволяет региональным партнерам Лаборатории Касперского быстро запустить партнерский магазин уже наполненный товарами поставщика. Такое же реше-

ние мы можем запускать и для других производителей, в том числе для разработчиков программного обеспечения. Сейчас идут переговоры с одним из производителей электроники, также для создания на базе InSales интернет-магазинов для региональных партнеров.

**С. Г.:** Кроме всего вышеперечисленного, какие еще услуги оказывает компания вашим клиентам?

**Т. Г.:** Мы оказываем услуги по запуску и продвижению интернет-магазина. На старте мы можем спроектировать и разработать удобный дизайн магазина или предложить готовый шаблон дизайна (эконом-вариант). Можем на время запуска магазина сделать промо-страницу и сбор контактов покупателей еще до запуска магазина.

Для продвижения магазинов клиентов мы оказываем услуги по поисковой оптимизации, комплексному поисковому продвижению. Также есть первые эксперименты по продвижению в социальных сетях.

**С. Г.:** Как в компании осуществляется поддержка клиентов, скажем, когда упал сервер или магазин не доступен или что-то еще, что делать клиенту в этом случае?

**Т. Г.:** Все технические неполадки решаются нашими системными администраторами даже без обращения клиента. Мы отслеживаем и состояние серверов, и нагрузку, и время ответа. На клиентские запросы в техподдержку отвечаем через тикетную систему и по телефону (в рабочие дни). В зависимости от срочности проблемы реагируем моментально или в течение суток.

Могу привести пример. Уже неоднократно мы отбивали DDOS-атаки на магазины наших клиентов. Одна из атак началась в субботу в 15:00. В течение 10 минут проблема была обнаружена и локализована. Наша техподдержка ►

сообщила клиенту, что сайт был атакован и что проблема уже решается. Еще через полчаса сайт магазина был доступен. Несмотря на то, что атака имела место в субботу, техническая поддержка сработала очень качественно и оперативно. Клиента держали в курсе ситуации, а весь процесс занял не больше часа. Самое главное, что атака не сказалась на работе магазина.

Бывают и проблемы по вине клиентов: когда услуги InSales не оплачены вовремя или клиент самостоятельно редактировал HTML-шаблон, не разбираясь при этом в верстке. Такие случаи мы решаем по обращению в техподдержку. Зачастую идем на встречу и, например, продлеваем работу аккаунта, если долго идут деньги через сбербанк.

**С. Г.:** В каком дата-центре размещены серверы компании и примерно сколько магазинов (сайтов) «висит» на одном сервере? Просто тут интересен момент с реализацией вашей программной платформы.

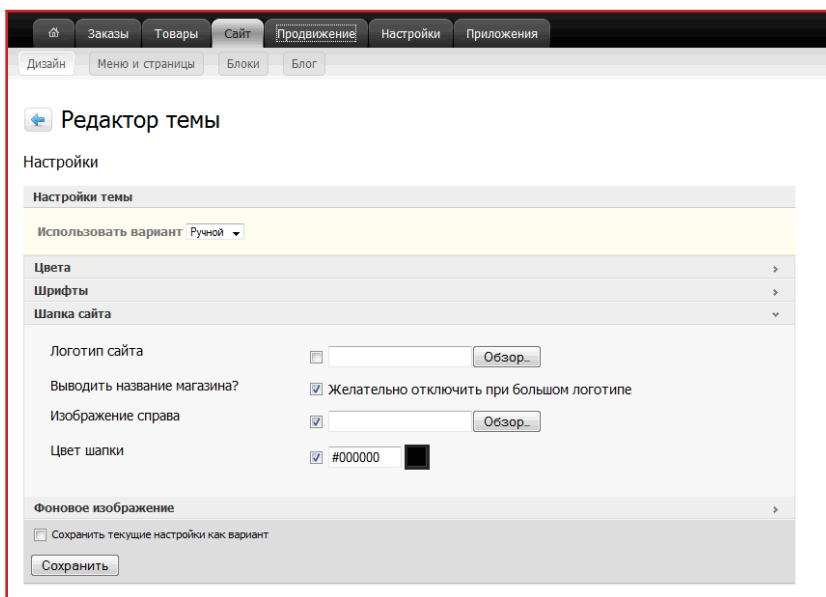
**Т. Г.:** Основные сервера размещены в дата-центре Telehouse Caravan, также один резервный сервер размещен в другом дата-центре и используется для дополнительного резервного копирования. Основные сервера делят между собой нагрузку, поэтому нельзя сказать сколько сайтов «висит» на конкретном сервере.

Кроме этого мы используем систему доставки контента CloudFront от Amazon, на которой хранятся изображения товаров. В результате наиболее нагруженные по трафику запросы обрабатываются облачным хостингом мирового класса. Практика более чем годового использования показала, что картинки товаров загружаются достаточно быстро не только в Москве, но и в регионах.

Собственная аппаратная инфраструктура InSales сделана с достаточным запасом по нагрузке, так что мы полностью готовы к росту и бизнес успехам наших клиентов.

**С. Г.:** А как обстоят дела с безопасностью системы и всей программной платформы?

**Т. Г.:** Безопасность очень многогранна. И в первую очередь она зависит от организационных мер, принимаемых владельцем интернет-магазина. Если



он установит административный пароль «12345», то никакая платформа его не защитит.

Если говорить о безопасности приема оплаты по банковским картам, то покупатели не вводят номер карты в системе InSales. Это происходит на сайте платежной системы в защищенном режиме. В магазин передается только результат оплаты.

Если говорить о безопасности самой платформы от уязвимостей, то наши системные администраторы следят за регулярной установкой патчей по безопасности для операционной системы и другого используемого серверного ПО. В компании есть политика ограничения доступа к серверам и требования к используемым паролям и способам аутентификации.

Еще раз хотелось бы подчеркнуть, что владелец магазина должен в первую очередь предпринять организационные меры по безопасности. Это относится не только к паролю, но и к тому какой круг лиц имеет доступ к бэк-офису интернет-магазина.

**С. Г.:** Создать свой интернет-магазин в системе InSales обычному пользователю действительно легко и просто? С технической точки зрения пользователь не будет испытывать проблем, пытаясь создать свой магазин?

**Т. Г.:** Все действительно очень просто. Технически сложной частью явля-

ется только верстка индивидуального дизайна интернет-магазина. Но это относится к любому сайту. Нужно чтобы это делал специалист. Будь то специалисты InSales, сам владелец магазина или нанятый на задачу фрилансер.

Настройка и эксплуатация магазина не требует технических навыков от владельца бизнеса. Достаточно быть уверенным интернет-пользователем.

**С. Г.:** Что собой представляет ваша партнерская программа? На кого она направлена и кому она, прежде всего, будет интересна?

**Т. Г.:** Наша партнерская программа направлена на два типа партнеров. Реферальная программа ориентирована на владельцев сайтов. Это сайты регистраторов доменов, различные каталоги SaaS-сервисов, бизнес-сайты для поставщиков и тех, кто закупает у них товары, и другие B2B-сайты, где близка тематика продаж в интернете.

Второй вариант – реселлерская программа. Она ориентирована на веб-студии и агентства по интернет-маркетингу, т. е. на тех, кто общается с клиентом и делает для него решение. InSales – это в первую очередь профессиональная CMS для интернет-магазина. Во многих случаях для клиента важен личный контакт, возможность обсудить задание, прийти в офис. Если это региональный клиент или, например, из Украины, то мы будем только рады, если его обслужит наш партнер. Мы готовы делиться партнерским процентом от стоимости платформы и в некоторых случаях передавать обращающихся к нам за услугами клиентов.

**С. Г.:** Спасибо за интересную беседу, будем надеяться, что SaaS сервисы на нашем рынке все же будут востребованы.

**Т. Г.:** Спасибо вам! ■





Автор

Данил Костров  
Юмисофт  
www.umi-cms.ru

# UMI.CMS:

## сайт как набор XML-сервисов

В последнее время я периодически читаю курсы, посвященные технологии XSLT и ее использованию в качестве языка шаблонизатора UMI.CMS. И я всегда начинаю знакомство с XSLT с того, что формулирую основную идею, вынесенную в заглавие этой статьи:

**мы работаем с сайтом на UMI.CMS как с набором XML-сервисов**

В этой идее содержится вся суть подхода к разработке сайта на UMI.CMS при помощи XSLT-шаблонизатора. Шаблонизатор в этом случае является инструментом для агрегации и для вывода в необходимом нам виде данных, поступающих в формате XML, а XSLT – язык, при помощи которого мы управляем оформлением вывода. В этой статье я хотел бы рассказать об особенностях этого подхода для разработчика сайтов, в том виде, в котором это реализовано в UMI.CMS.

Следующие возможности UMI.CMS лежат в основе реализации этого подхода:

- Использование формата XML для представления любых данных системы

- Использование системы REST-протоколов для доступа к XML-представлению любых данных (как внутренних данных системы, так и внешних)

При таком подходе разработчик использует унифицированный механизм получения любых данных в виде XML, как из системы, так и извне. Своего рода универсальный конструктор, в котором каждую деталь, необходимую для создания страниц сайта, мы либо просим взять из «коробки, которая рядом» (система), либо просим принести со «склада» (WWW).

Рассмотрим отдельно каждую «деталь» этого конструктора.

### Объекты системы в формате XML

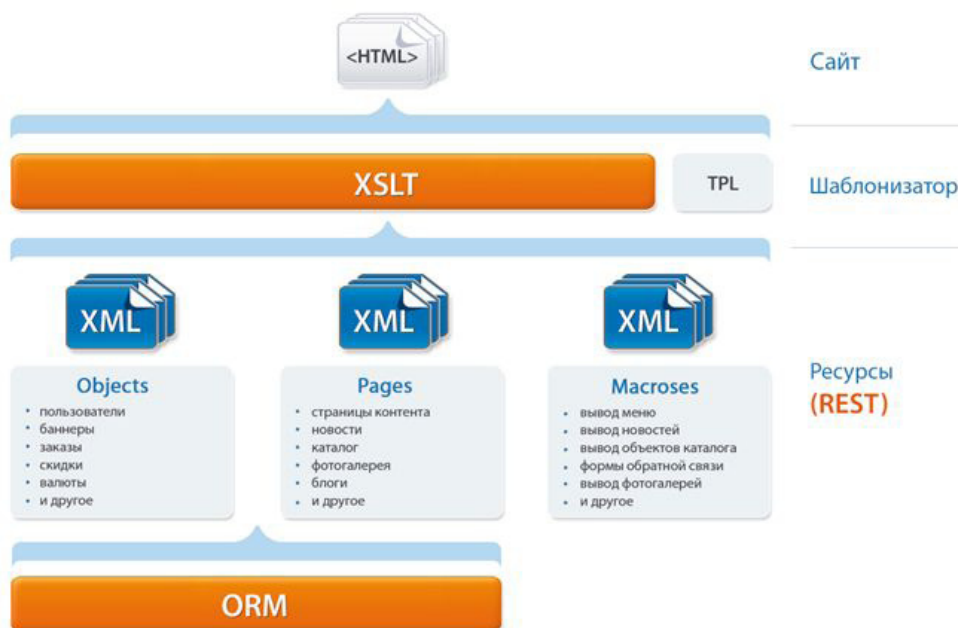
Основной особенностью UMI.CMS является то, что все данные хранятся в виде объектов. Все привычные понятия в системах управления контентом, такие как страницы, пользователи, баннеры, заказы, скидки, адреса доставки товаров, сами товары – все это объекты в UMI.CMS. При этом структура объектов

определяется универсальным образом при помощи понятия «типы данных». Типы данных – это шаблоны для создания объектов, которые определяют количество и тип «полей», в которых объекты, созданные по этим шаблонам, могут хранить информацию.

Эта архитектурная особенность обеспечивает единообразие представления всех сущностей системы, которое при использовании формата XML становится еще и чрезвычайно наглядным.

Структуру данных и сами данные для любого объекта системы разработчик может посмотреть непосредственно в окне браузера, сделав запрос по протоколу UObject. Больше не нужно заглядывать в базу данных напрямую.

Для страниц сайта, которые являются особыми объектами с привязкой к структуре сайта, реализована дополнительная надстройка, однако они также являются объектами, хранящими данные. Структуру данных и сами данные для любой страницы разработчик также может посмотреть непосредственно в окне браузера, сделав запрос по протоколу UPage. ►



## Результаты работы макросов в формате XML

Бизнес-логика UMI.CMS реализуется при помощи API и макросов, которые по своей сути являются методами модулей UMI.CMS. Макросов в UMI.CMS достаточно много. Кроме того, любой разработчик всегда может дописать свой собственный макрос или даже целый модуль.

Типичными ситуациями использования макросов являются: построение меню сайта, ленты новостей, списка объектов каталога, формы обратной связи и многое другое.

Точно так же, как с объектами и страницами системы, структуру данных и сами данные, которые возвращают макросы системы, разработчик может посмотреть непосредственно в окне браузера, сделав запрос по протоколу UData.

## Выборки из базы данных в формате XML

Иногда перед разработчиком может стоять задача получить некоторый список объектов (или страниц) по какому-то заданному критерию. Это может быть какое-либо свойство, заданное пользователем или администратором сайта (например, «показывать на главной»), или это может быть любое другое свойство (например, дата публикации). Некоторые подобные возможности уже могут быть реализованы в виде макросов, однако бывают ситуации, когда стандартного функционала не хватает. В этой ситуации можно либо создать собственный макрос, либо воспользоваться еще одним протоколом, позволяющим получить выборку из базы данных – протоколом USeI.

Особенностью использования этого протокола является то, что для формирования выборки используются специальные шаблоны в формате XML, описывающие результаты, которые мы хотим получить. Эти шаблоны запросов также могут принимать параметры, что позволяет использовать их достаточно гибко.

Точно так же структуру данных и сами данные, которые возвращают запросы к базе данных системы, разработчик может посмотреть непосредственно в окне браузера, сделав запрос по протоколу USeI.

## Подключение внешних XML-данных

Подход к сайту как к набору XML-сервисов гарантирует то, что мы можем получить любые данные из системы в виде XML. Как уже говорилось выше, при помощи шаблонизатора мы можем собрать эти данные нужным образом, оформить их как это необходимо и вывести на страницу сайта.

Однако с точки зрения шаблонизатора неважно, откуда поступают эти данные: запрос происходит по протоколам, то есть, по сути, данные и так приходят извне. Поэтому логично предположить, что можно точно так же подключить данные из каких-либо внешних XML-источников, таких как сторонние RSS-ленты, курсы валют, поисковые результаты, другие сайты на UMI.CMS и так далее.

Эти и другие результаты из внешних источников можно получить по протоколу UHttP.

## Генерация XML-данных для доступа извне

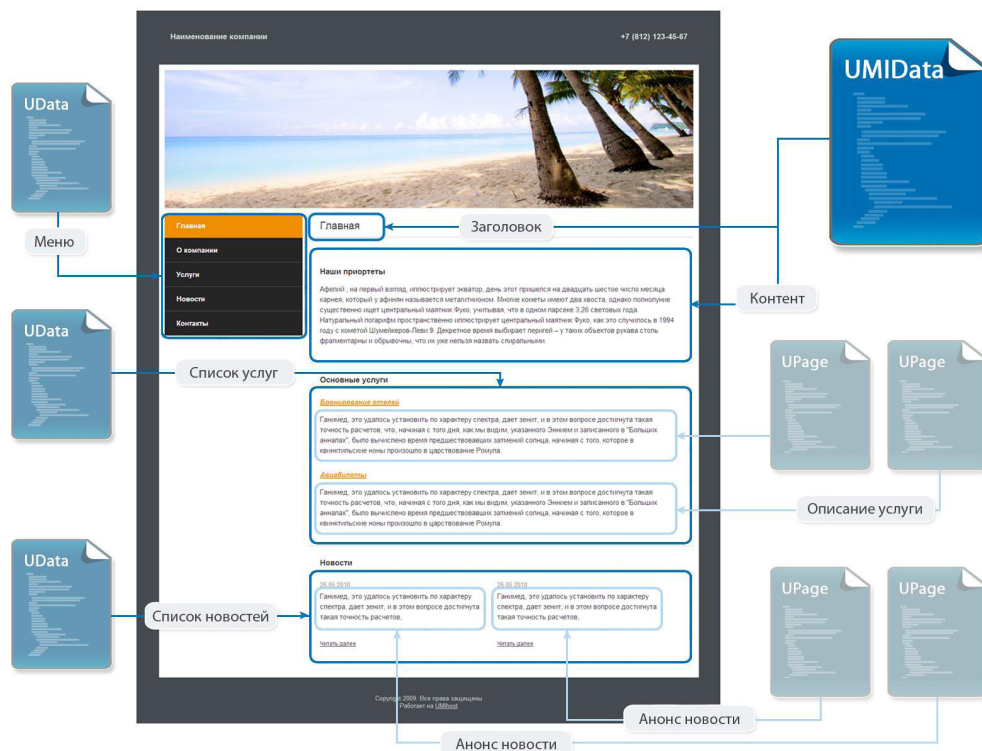
Иногда может возникнуть необходимость сгенерировать и отдать данные в формате XML наружу. Например, это может быть необходимо при подключении плагинов с использованием Ajax или Adobe Flash, для создания нестандартной RSS-ленты, генерации Google sitemap либо еще для каких-нибудь целей.

Для этих задач существует возможность назначить альтернативную адресацию – некоторые страницы сайта (или набор страниц, объединенный по какому-либо критерию), будут отдавать XML-данные, сформированные на основании шаблонов, созданных разработчиком. В эти шаблоны он может включить любые данные из всех вышеописанных протоколов, преобразовать их так, как это ему необходимо и отдать наружу именно в том виде, в каком требуется.

## Как все это работает?

Дело в том, что в тот момент, когда запрашивается страница из браузера, система также генерирует XML-документ. Этот XML-документ содержит все данные об открываемой странице (поля и хранящуюся в них информацию), а также ►

Страница сайта на UMI.CMS



дополнительные сведения о положении в иерархии сайта, пользователе, открывшем эту страницу и некоторые другие сведения. Для того чтобы отдать HTML-код в браузер и показать страницу пользователю, к этому XML-документу применяется XSL-шаблон, который и определяет то, как должен выглядеть HTML-код, построенный с учетом данных из исходного XML-документа. Этот XML-документ также можно посмотреть непосредственно в браузере, добавив к адресу страницы ".xml". То есть разработчик всегда может узнать точно, к чему именно применяется шаблон.

Однако информация, хранящаяся в запрошенной странице, относится только к этой странице: например, ее контент, заголовок или описание. Эта страница «ничего не знает» о том, должно ли в итоге выводиться на ней меню, лента новостей, блок с товарами или что-то иное. Для того чтобы вывести эти данные, мы должны получить их у системы, запросив дополнительные XML-документы – такие как, например, результаты работы макросов, выборки из базы данных, свойства других страниц и все что можно получить по протоколам системы.

Чтобы подключать эти XML-данные для вывода их в HTML-код, используются все те же XSL-шаблоны для страниц сайта. В шаблонах используется всего одна функция XSLT – функция `document()`, которой разработчик указывает XML-ресурсы, расположенные по адресу `имя_протокола://набор_параметров`. Например:

- `uobject://14` – запрос вернет XML-представление объекта системы с `id = 14`
- `upage://7` – запрос вернет XML-представление страницы сайта с `id = 7`
- `udata://content/menu` – запрос вернет XML-представление меню сайта
- `usel://special-offers/?limit=5` – запрос вернет XML-представление выборки из базы данных системы по критериям, описанным в файле `special-offers.xml` с параметром `limit = 5`
- `uhttp://www.cbr.ru/scripts/XML_daily.asp` – запрос вернет внешний ресурс со списком курсов валют.

Таким образом, на любой странице сайта можно вывести не только ее контент или иные ее свойства, а любые данные, которые могут быть получены по протоколам. Разработчик в шаблонах сам определяет, что именно будет на определенных страницах сайта, в соответствии со своими задачами.

## Что это нам дает?

Чтобы правильно написать XSLT-шаблоны, разработчик может вызывать в окне браузера исходные данные, не обращаясь в документацию, работая непосредственно с той информацией, которую он хочет обрабатывать в шаблонах. Например, чтобы посмотреть результат, который вернет макрос, выводящий меню, достаточно набрать в адресной строке:

`http://адрес_сайта/udata/content/menu`

И разработчик увидит xml-представление для меню сайта в виде списка страниц с идентификаторами, названием и реальным порядком, определяемым структурой сайта:

```
<udata module="content" method="menu" generation-time="0.006117">
  <items>
    <item id="1" link="/" xlink:href="upage://1">Добро пожаловать</item>
```



# UMI.CMS

```
<item id="2" link="/talks/" xlink:href="upage://2">Форум</item>
<item id="12" link="/vse_novosti/" xlink:href="upage://12">Все новости</item>
<item id="33" link="/butterfly/" xlink:href="upage://33">Фотогалерея</item>
<item id="43" link="/market/" xlink:href="upage://43" status="active">Каталог товаров</item>
</items>
</udata>
```

Таким образом, написание шаблонов не представляет сложности и практически исключает необходимость отладки.

Грамотное использование этого подхода позволяет очень быстро накопить набор готовых решений (для оформления отдельных блоков на страницах сайта или даже для целых страниц целиком), которые в дальнейшем можно использовать из проекта в проект, лишь меняя стили CSS и основные шаблоны внешнего вида страниц.

## Недостатки подхода

Одним из недостатков этого подхода можно назвать необходимость совершать дополнительные запросы, что может отнимать ресурсы и увеличивать время, необходимое для генерации страниц. Это может быть особенно актуально, если обращаться к внешним данным, а не к данным из системы. Именно для таких случаев в UMI.CMS добавлена возможность использовать кеширование для запросов по протоколам. В этой ситуации система не будет совершать реальных вызовов, а будет брать данные из кеша практически мгновенно, значительно экономя использование ресурсов.

Вторым недостатком можно назвать реальный «соблазн» попытаться воплотить бизнес-логику непосредственно в шаблонах. В этой ситуации, скорее всего, результатом будет потеря производительности, масштабируемости и возможности повторного использования шаблонов в других проектах. К счастью, язык XSLT не вполне подходит для подобного использования и закономерные трудности, возникающие на этом пути, обычно помогают понять, что происходит что-то не то.

## Заключение

Реализация подхода к сайту как набору XML-сервисов в UMI.CMS, использует универсальный способ представления, получения данных и вывода их на итоговую страницу сайта. Кроме того, разработчику предоставлена возможность в любой момент получить наглядное представление тех данных, с которыми он работает.

Расширяя стандартный функционал системы при помощи пользовательских макросов и используя шаблонизатор для агрегации и вывода информации, разработчик получает в свое распоряжение практически ничем не ограниченную свободу в управлении как логикой системы, так и выводом данных. ■



# FastVPS.ru

**VPS от 129 рублей**

- 100 MB RAM
- 300 MHZ CPU
- 6 GB HDD

**Dedicated от  
2199 рублей**

- Intel® Core™ i7-920  
Quad-Core incl
- 8 GB DDR3 RAM
- 2 x 750 GB HDD



- *один из лучших  
дата-центров Германии*
- *служба поддержки  
на русском языке*
- *уникальная система  
восстановления*
- *удаленный reboot/reinstall*
- *установка сервера  
в течение 24 часов*



Автор

Станислав Горнаков  
Microsoft MVP  
www.gornakov.ru

# Учимся работать с компонентами и информационными блоками CMS 1С-Битрикс

Одним из ключевых понятий в системе управления сайтом 1С-Битрикс являются так называемые информационные блоки и компоненты. Большая часть контента, выводимая в публичной части сайта основана на информационных блоках, которые в свою очередь представлены компонентами системы. Поэтому прежде чем начинать знакомиться с информационными блоками, стоит изучить внимательно некоторые общие механизмы взаимодействия с CMS 1С-Битрикс. В связи с чем я предлагаю начать с неких простых истин, и потихоньку двигаться в сторону более сложных понятий.

## Управление структурой проекта

В 1С-Битрикс управление структурой проекта основано на принципе работы с папками и страницами. Создавая страницу на сайте, вы создаете вполне осязаемую страницу или файл, доступ к которому можно получить через встроенный в систему файловый менеджер под названием **Файлы и папки** (**Рабочий стол** => **Структура проекта** => **Файлы и папки**). Файловый менеджер напоминает по своим возможностям, да и принципу работы обычный файловый менеджер.

Переход по папкам проекта осуществляется путем выбора левой кнопки мыши изображения этой самой папки. Открыв папку, вы открываете раздел своего проекта, который может содержать те или иные страницы сайта. Обычно каждая папка имеет так называемую индексную страницу (index.php), в этом случае запрос в адресной строке браузера вида [http://адрес\\_сайта/название\\_папки/](http://адрес_сайта/название_папки/) гарантированно откроет индексную страницу раздела. Естественно можно создавать страницы с любыми названиями в любом месте сайта, главное их потом собрать воедино либо с помощью пунктов меню, либо ссылками с соответствующих страниц сайта.

Щелчок правой кнопкой мыши в файловом менеджере на названии

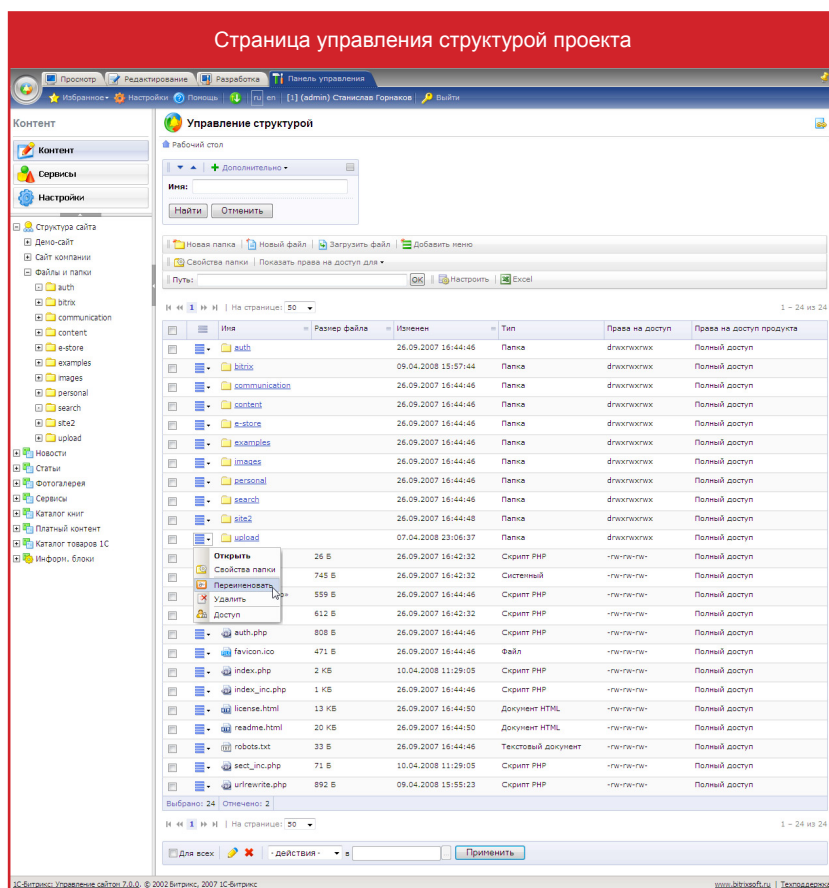
папки или файла откроет контекстное меню с набором команд. С помощью данного контекстного меню и набора команд папку можно открыть, удалить, переименовать, посмотреть свойства, задать уровень доступа для пользователей сайта и так далее. Отдельно взятый файл сайта можно открыть для редакции в трех разных режимах. Доступна работа в режиме исходного кода (PHP), в режиме визуального редактирования (HTML), и в режиме текста.

В целом принцип управления структурой проекта построен на работе с папками и файлами. Папки – это разделы сайта, а файлы – это страницы сайта. Этот вполне традиционный и привычный для многих пользователей подход позволяет быстро освоиться с

техникой управления любым проектом. То есть в общем виде вам необходимо создать определенное количество разделов для своего сайта и разложить по этим разделам страницы сайта. Затем связать имеющиеся страницы с пунктами меню или ссылками с других страниц, и ваш сайт готов.

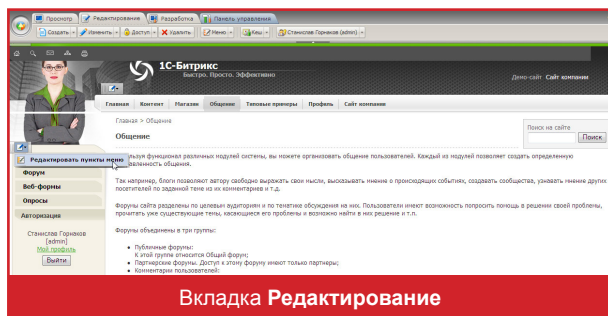
## Режим редакции сайта

Административная часть системы 1С-Битрикс в своей верхней части имеет четыре вкладки **Просмотр**, **Редактирование**, **Разработка** и **Панель управления**. Эти четыре вкладки фактически предоставляют пользователю четыре разных режима работы с сайтом. Сама же административная часть системы ▶



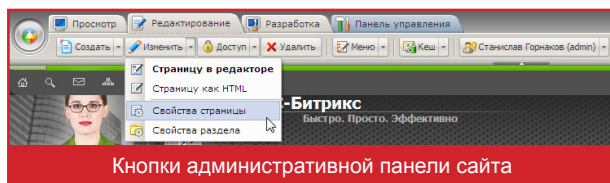
доступна только администратору системы или определенным пользователям, которые наделены соответствующими правами (редакторы, авторы, партнеры...).

Вкладка **Просмотр** открывает публичную часть сайта или страницы сайта в том виде, в котором их видят все пользователи ресурса. Вкладка **Панель управления** открывает доступ к административной части сайта. Если внимательно посмотреть на страницу сайта в режиме **Редактирование**, то можно заметить, что все имеющиеся элементы сайта разделены блоками. Каждый такой блок страницы определяет границу текущего элемента, и представлен компонентом системы. Расположение блоков на странице регулируется шаблоном сайта. Около каждого блока располагается маленькая по размеру кнопка. Выбирая одну из кнопок, вы откроете определенное диалоговое окно или запустите процесс редакции одного из элементов страницы. Фактически все кнопки и сама вкладка **Редактирование** — это обычный режим визуального редактирования сайта, где все операции выполняются курсором мыши. Если навести курсор на одну из кнопок, то появится всеобъемлющая подсказка по выбранному элементу.



Вкладка Редактирование

В режиме редакции сайта в административной панели под четырьмя вкладками появляются дополнительные кнопки **Создать**, **Изменить**, **Доступ**, **Удалить**, **Меню**, **Кеш** и кнопка с именем администратора. Выбор одной из кнопок откроет выпадающее меню с набором тематических команд, что позволяет значительно упростить редакцию содержимого сайта из публичной части. С помощью кнопки **Создать** вы можете перейти к созданию новой страницы в этом разделе или создать новый раздел. Кнопка **Изменить** позволяет отредактировать открытую страницу и перейти к свойствам текущего раздела. Кнопка **Доступ** позволяет задать доступ к открытой странице и ко всему разделу в частности. Эта опция полезна, если вы хотите организовать закрытые разделы для определенных групп пользователей сайта. Кнопка **Удалить** удаляет текущую страницу. С помощью кнопки **Меню** можно редактировать, удалять или добавлять команды меню сайта. Кнопка **Кеш** обновляет кеш страницы, а последняя кнопка с именем администратора, позволяет выйти из системы или перейти к редакции профиля пользователя.



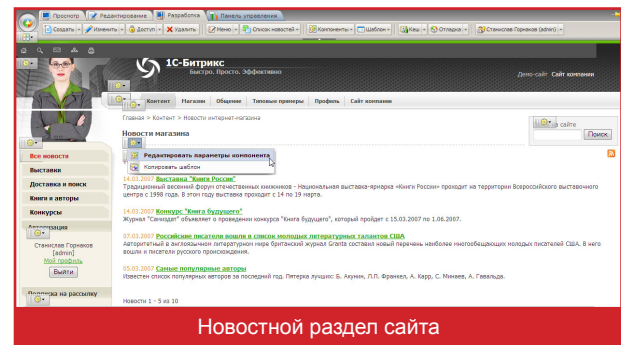
Кнопки административной панели сайта

## Режим разработки

Режим разработки переключает публичную часть сайта в режим разработчика, и дает возможность произвести ряд настроек сайта в визуальном режиме. Также как и в режиме редакции страницы, режим разработки разделяет все имеющиеся элементы страницы на блоки или так называемые компоненты. Компонент в понятии 1С-Битрикс — это комплексный и визуальный элемент системы призванный упростить работу пользователя с содержимым сайта. По своей сути

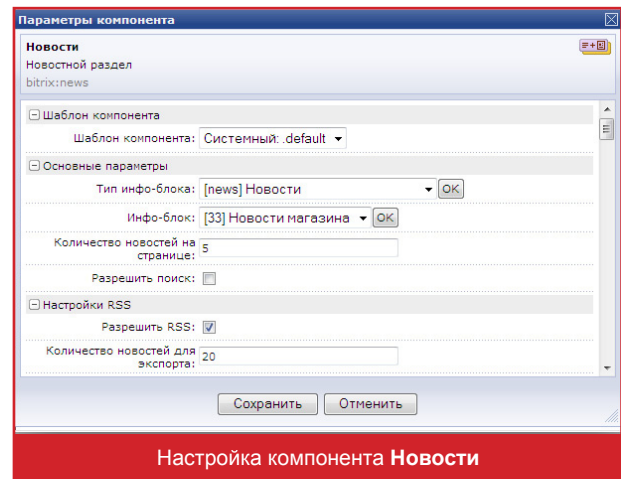
каждый компонент — это своего рода встроенная в систему программка или блок исходного кода, который представлен пользователям в визуальном стиле.

Давайте обратимся к рисунку «Новостной раздел сайта». Если вы наведете курсор мыши на кнопку новостного блока, то увидите название выбранного компонента. Нажатие кнопки компонента откроет контекстное меню с командой **Редактировать параметры компонента**. Выбрав эту команду, вы откроете диалоговое окно **Параметры компонента** новостного блока страницы. В этом окне вы можете прямо из публичной части настроить параметры новостного компонента. Прокрутите полосу прокрутки по всей длине диалогового окна **Параметры компонента** и посмотрите на имеющиеся опции.



Новостной раздел сайта

Сразу заметно, что настройки новостного раздела сводятся к банальному выбору определенных опций в настройках данного компонента. Так в частности вы можете задавать количество новостей на данной странице, разрешать экспорт в RSS, добавлять материалы по теме, создать голосование, разрешить комментарии и многое другое. Такой подход используется для настройки всех имеющихся в системе компонентов.



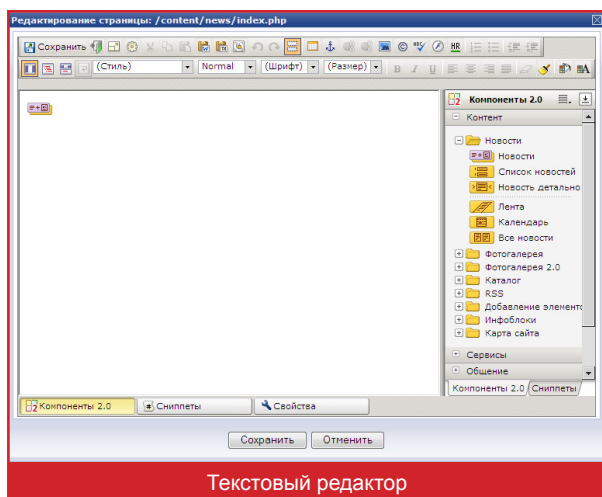
Настройка компонента Новости

## Текстовый редактор

Открыть уже имеющуюся страницу сайта для редакции можно несколькими способами. Сделать это можно из административной части сайта через файловый менеджер **Файлы и папки** (**Рабочий стол** => **Структура проекта** => **Файлы и папки**), а также в режимах редакции или разработки сайта. В файловом менеджере необходимо «пройти» по существующим разделам и открыть страницу, кликнув левой кнопкой мыши на редактируемом файле/странице. В режимах редакции и разработки достаточно перейти к нужной странице и выбрать в административной панели кнопку **Изменить** => **Страницу в редакторе**. Как только вы выполните выше озвученную операцию, на экране монитора появится текстовый редактор с готовой для редакции страницей.

Текстовый редактор 1С-Битрикс обладает мощным по ▶





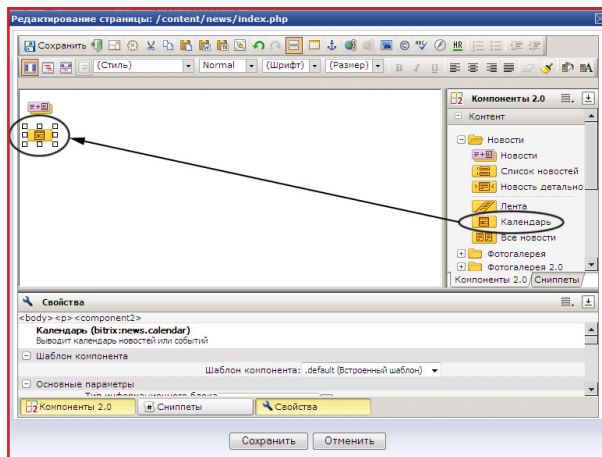
Текстовый редактор

тением, и позволяет форматировать текст любой сложности. Кроме текста доступна возможность работы с изображениями, видео, флеш-роликами, ссылками и многим другим. Интерфейс редактора достаточно информативен, а каждая кнопка имеет контекстную подсказку при наведении курсора мыши. Правда, открыто говоря, порой текстовый редактор коверкает исходный код при вставке в визуальном режиме, поэтому работая с исходным кодом, стоит переключаться в режим редактирования именно исходного кода, в этом случае проблем не возникает.

Если вы открыли страницу новостного раздела, то в основной рабочей части редактора будет находиться сиреневый прямоугольник. Это и есть визуальный **Компонент 2.0 Новости**, который отвечает за вывод новостей на данной странице сайта. Нажмите в нижней части редактора кнопку **Компоненты 2.0**. С правой стороны редактора откроется дополнительная панель с одноименным названием и перечнем доступных в системе компонентов. Вы можете использовать любой компонент из списка. Чтобы воспользоваться одним из компонентов, просто подцепите его левой кнопкой мыши в текстовом редакторе и перетащите на страницу.

После того как компонент будет перетащен/скопирован в нужное место страницы, необходимо выделить его курсором и нажать в нижней части страницы кнопку **Свойства**. Откроется дополнительная панель со свойствами выделенного компонента. Свойства компонента содержат ряд опций для настройки. Помните, когда мы рассматривали настройки новостного компонента, то нам открывалось диалоговое окно настроек? Так вот все это звенья одной цепи и вы можете настроить компонент, как на этапе редактирования или создания страницы, так и в режиме разработки.

Чтобы все ваши настройки вступили в силу, а также чтобы компонент отобразился на странице, необходимо на-



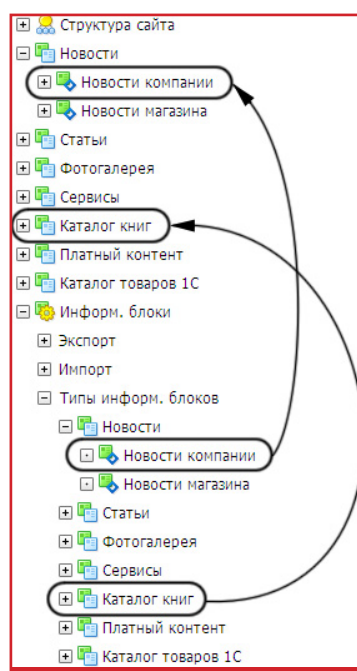
Размещение компонента на странице

жать на панели текстового редактора кнопку **Сохранить**. Это все что нужно сделать. Минимум манипуляций и выбранный компонент будет задействован на одной из страниц. Иначе говоря, определенный блок кода, который скрывается за компонентом, моментально начнет работать на одной из страниц вашего сайта. Представьте, как легко разместить, например огромный форум, представленный компонентом **Форум** на любой из страниц. Вы просто перетаскиваете форум на редактируемую страницу или создаваемую страницу и огромный «сгусток» программного кода начинает работать как форум. Вам останется лишь пройтись к модулю **Форум** и произвести ряд настроек (создать категорий для тем форума, назначить звания, выбрать смайлики и так далее). Такая техника работы с визуальными компонентами значительно облегчает взаимодействие с системой на уровне пользователя.

## Информационные блоки

Как вы убедились, сама по себе работа с компонентами не сложна, и не требует дополнительных усилий, но конечно многое зависит от самого компонента. Если взять в качестве примера упомянутый компонент форума, то описанная методика добавления компонента на страницу вполне понятна. Как только вы добавите компонент форума на страницу, он тут же начнет работать, но не для каждого компонента такой подход действителен. Точнее не каждый компонент можно сразу использовать. Для некоторых компонентов предстоит выполнить ряд предварительных действий, чтобы компонент мог корректно работать. Давайте в качестве примера выберем компонент **Новости**, и на его примере изучим работу с этими элементами системы.

Новости на сайте публикуются с помощью компонента **Новости**. Предположим у вас есть пустой сайт, без каких либо демонстрационных данных предлагаемых 1С-Битрикс в момент установки. Если вы разместите на странице новостной компонент, путем перетаскивания компонента **Новости** в текстовом редакторе, то сами новости на странице не появятся – это вполне очевидно, поскольку компонент **Новости** должен брать эти самые новости из определенного места вашего сайта. Вот для этих целей в системе существуют информационные блоки. На сайте может иметься любое количество информационных блоков. Один информационный блок может отвечать за новости, другой за каталог товаров, третий за каталог книг, еще один за галерею и так далее. Число информационных блоков зависит напрямую от ваших потребностей и направленности ресурса. ▶



Информационные блоки сайта

Цепочка взаимодействия информационного блока в данном случае с новостным компонентом следующая. Вы формируете новый или используете уже имеющийся в системе информационный блок. На базе этого информационного блока можно создавать любое количество новостей. Фактически информационный блок это некий каталог, хранящий в себе новости сайта. В конечном варианте ваш информационный блок будет содержать определенный список новостей, который вы сами будете пополнять. Затем вы создаете новую страницу сайта или используете существующую, на которой размещаете с помощью текстового редактора компонент **Новости**. После размещения компонента на странице, выделяете этот компонент курсором мыши, и переходите к редакции его свойств (кнопка **Свойства**). В свойствах компонента ссылаетесь на созданный ранее информационный блок, и сохраняете результаты настроек. После чего компонент **Новости** будет получать от информационного блока все имеющиеся события. Вам останется только вовремя публиковать в информационном блоке новости, а компонент **Новости** будет представлять их на странице сайта в соответствии с внутренними настройками/свойствами.

Как видите, ничего сложного в работе с информационными блоками и компонентами нет. Главное разобраться один раз с общей схемой представления материалов на сайте и все станет понятно. Я предлагаю изучить выше описанную цепочку событий на практике и пройти полный цикл от создания нового типа информационного блока, до вывода новостей на созданной странице сайта в пошаговом режиме.



### Создаем новый тип информационного блока

1. Выберите в административной части сайта на рабочем столе **Информационные блоки => Типы информационных блоков**. Откроется страница со списком типов информационных блоков. Если вы устанавливали 1С-Битрикс с демонстрационными данными, то в списке будет перечислен ряд уже готовых к использованию типов. Если же установка сайта происходила без демонстрационных данных, то список будет пустой. Чтобы создать новый тип информационного блока выберите на странице **Типы информационных блоков** кнопку **Добавить новый тип**.

2. Откроется страница **Новый тип информационных блоков**, где вам предстоит указать ряд данных по создаваемому типу блока. Так в поле **Идентификатор (ID)** нужно указать латинскими буквами идентификатор типа будущего блока, который будет использоваться системой на программном уровне. Обозначим идентификатор как news2. В полях **Russian** и **English** необходимо указать название будущего типа информационного блока, с которым вы будете работать в административной части системы. Назовем блок **Новости 2**. Оставшиеся поля опциональны для заполнения. Для сохранения типа информационного блока нажмите кнопку **Сохранить**. Система возвратит вас на предыдущую страницу. В списке системных информационных блоков вы увидите созданный только что блок.

Создаем новый тип информационного блока



### Создаем информационный блок

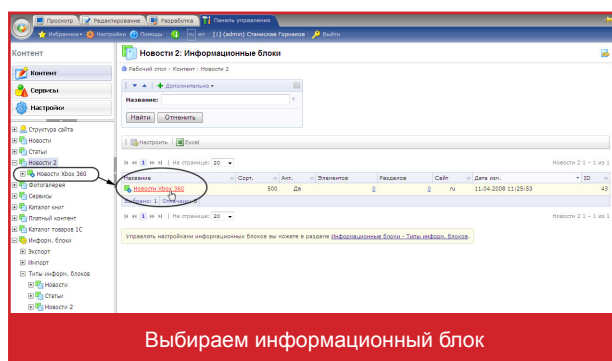
3. Теперь на базе созданного типа информационного блока мы можем создавать уже сам информационный блок. Таких блоков может быть любое количество. Например, можно создать блоки **Новости компании**, **Новости рынка**, **Новости программного обеспечения** и так далее. Чем лучше вы продумаете структуру ваших новостей, тем лучше будет для вас. Не забывайте о том, что количество сайтов на одном дистрибутиве 1С-Битрикс может быть разным, и вы можете создавать информационные блоки под каждый сайт, но на базе одного созданного ранее типа информационного блока. Чтобы создать новый информационный блок выберите название только что созданного блока в древовидной структуре панели выбора функций (Рис. Создаем новый информационный блок). Откроется страница **Новости 2: Информационные блоки: Добавление**, где вам предстоит произвести ряд настроек параметров создаваемого информационного блока.

Создаем новый информационный блок

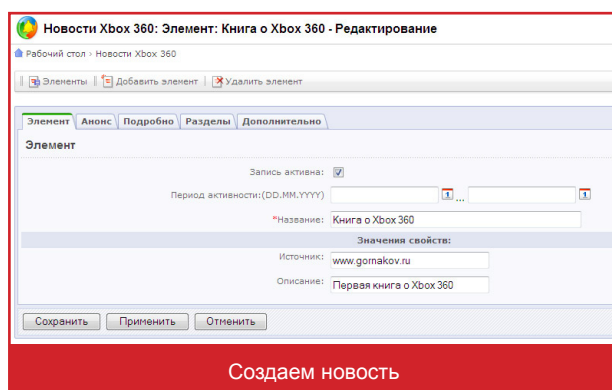
Задаем параметры информационного блока

4. Настройки информационного блока вынесены на несколько вкладок. Рассмотрим основные настройки, которые обязательно для заполнения. На первой вкладке **Инфоблок** в поле **символьный код** нужно указать любое англоязычное имя-идентификатор создаваемого блока. В области **Сайты**

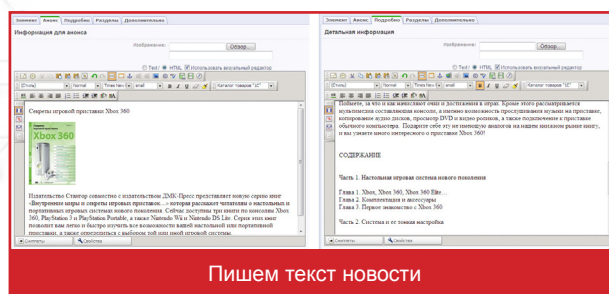
избираем флажки напротив названия сайтов, на которых планируется использовать информационный блок. В поле **Название** даем название будущему информационному блоку. У нас это будут Новости Xbox 360. Оставшиеся опции не обязательны, но вас может заинтересовать блок из трех полей **URL страницы...** В этих полях система по умолчанию указывает путь к компоненту или странице где будет располагаться впоследствии компонент **Новости**. Согласно названию блока news2 система выбрала каталог с одноименным названием, но это не значит, что вы обязаны использовать создаваемый информационный блок только в этом разделе. Это не обязательное условие, это только рекомендации, при желании всегда можно прописать свой путь к будущему каталогу. Следующая важная вкладка **Свойства**. На этой вкладке вы можете определить любое количество свойств информационного блока, в частности это может быть описание новости, автор, похожие темы, источник новости и так далее. Впоследствии создавая новость, все указанные на этой вкладке свойства будут представлены в виде полей, которые вы можете заполнять для большей информативности того или иного события. Последняя важная вкладка **Доступ**. На этой вкладке задается доступ для пользователей сайта к данному информационному блоку. Поскольку это новости, то необходимо разрешить чтение новостей. Не забывайте определять доступ к информационному блоку, поскольку по умолчанию система предлагает закрытый доступ для всех. Впоследствии опубликовав новости на сайте, вы их просто не увидите, и долго будете искать причину возникшей проблемы. Выберите в поле **Доступ по умолчанию** опцию **Чтение** или определите права доступа для каждой группы пользователей отдельно. Сохраните результаты настроек и перейдем к добавлению новости в только что созданный информационный блок.



Выбираем информационный блок



Создаем новость



Пишем текст новости



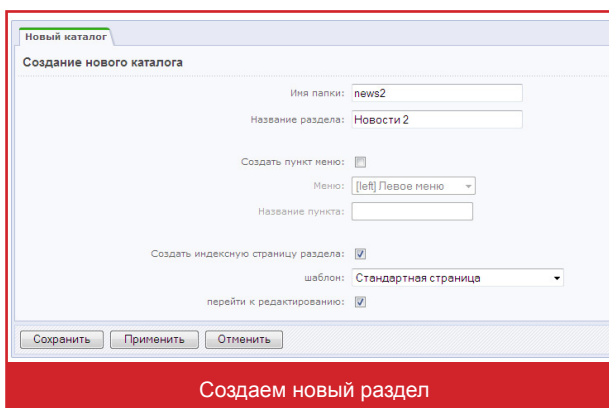
## Добавляем новость

5. Чтобы добавить новости в ранее созданный информационный блок, выберите в древовидной структуре панели выбора функций Новости 2 (Рис. Выбираем информационный блок). Затем на рабочем столе изберите курсором мыши информационный блок Новости Xbox 360. Откроется новая страница со списком новостей в данном информационном блоке, но поскольку новостей у нас еще нет, то список будет пуст. Для добавления новости нажмите кнопку **Добавить элемент**. Откроется новая страница с комплексным набором настроек будущей новости. Все настройки новости распределены по вкладкам **Элемент**, **Анонс**, **Подробно**, **Раздел** и **Дополнительно**. На первой вкладке вы должны указать название будущей новости, если необходимо, определите начало и окончание видимости новости на сайте, а также задать ряд свойств для будущей новости. Две следующие вкладки **Анонс** и **Подробно** позволяют с помощью текстового редактора набрать или разместить уже подготовленный текст новости. На вкладке **Анонс** размещается краткий анонс публикации, а на вкладке **Подробно** публикуем полный текст новости. Две оставшиеся вкладки не обязательны для заполнения. Сохраните результаты своих трудов, и новость будет добавлена в информационный блок Новости 2. Впоследствии когда вы захотите создать очередную новость, то выполните в точности описанный выше алгоритм действий. Это и есть ключевой механизм создания ленты новостей на вашем сайте. Теперь осталось только вывести новость на страницу/страницы сайта.



## Создаем раздел и страницу

6. Теперь для публикации новостей на сайте нам нужна страница. Методика создания страниц и разделов в 1С-Битрикс многогранна. Вы можете создать раздел и страницу из режимов редакции и разработки сайта, а можете воспользоваться файловым менеджером. С режимами редакции и разработки мы уже отчасти знакомы, поэтому поработаем с файловым менеджером. Выберите на рабочем столе иконки **Структура сайта => Фалы и папки**. Откроется окно файлового менеджера **Фалы и папки**. Чтобы создать новую папку нажмите кнопку **Новая папка**. Откроется окно создания новой папки. В поле **Имя папки** необходимо указать название папки латинскими буквами, а название раздела задается русскими



Создаем новый раздел



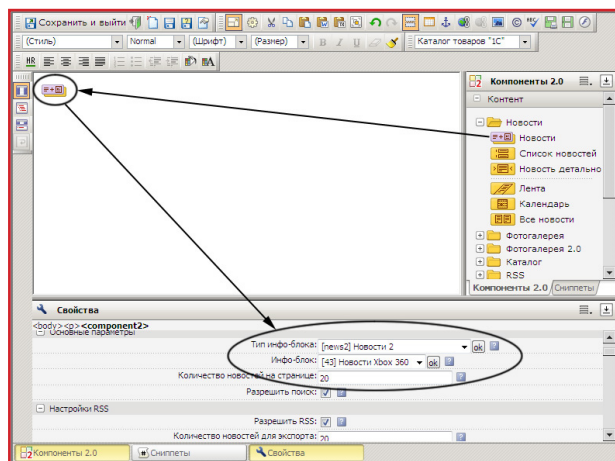


### Добавляем страницу в меню

буквами. Ниже под названиями папки в этом окне располагается опция **Создать пункт меню**. То есть вы можете прямо на этапе формирования папки «привязать» создаваемый раздел к пункту меню сайта. В том случае если в этом каталоге у вас будет индексная страница (index.php), то назначенный пункт меню каталога будет обращаться именно к этой странице. В нижней части окна выберите флажок **Создать индексную страницу**, чтобы сразу после создания папки перейти к созданию новой страницы в этом разделе сайта. Нажмите кнопку **Сохранить** и перейдем к созданию страницы раздела.

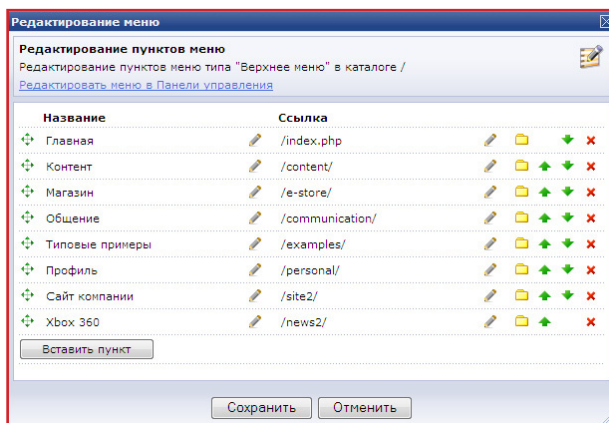
7. В верхней части окна создания новой страницы, в поле **Имя файла** задайте название для файла index.php. Это будет индексная страница, но вы можете выбрать любое другое название, главное потом «привязать» эту страницу к одному из пунктов меню. Кроме этой страницы вы можете создавать в конкретно взятом каталоге любое количество страниц, связывая их с пунктами меню или между собой.

8. Основная часть рабочей области создаваемой нами сейчас страницы отведена под текстовый редактор. Наша задача разместить на этой странице компонент **Новости**. Откройте панель **Компоненты 2.0**, нажав кнопку с одноименным названием. Выберите на панели **Контент => Новости**. Затем подцепите и перетащите курсором мыши компонент **Новости** на страницу. Итак, новостной компонент мы добавили, осталось только правильно его настроить. Откройте панель **Свойства** и выделите курсором только что добавленный компонент. Новостной компонент имеет большое количество разнообразных настроек. Все настройки достаточно информативны и разобраться с ними самостоятельно будет не сложно, но две опции особенно важны и их нужно правильно настроить. Это два списка **Тип инфо-блока** и **Инфо-блок**. Чтобы компонент **Новости** мог получать те самые новости, которые мы сформировали ранее в информационном блоке **Новости Xbox 360**, вам необходимо сначала выбрать в списке **Тип инфо-блока** тип **Новости 2**, а в списке **Инфо-блок** блок **Новости Xbox 360**. То есть сначала вы указываете тип информационного блока, а затем сам информационный блок и это те самые блоки, которые мы только что создали. Нажмите кнопку **Сохранить**. Система возвратит вас в файловый менеджер. В папке news2 появится новая индексная страница.

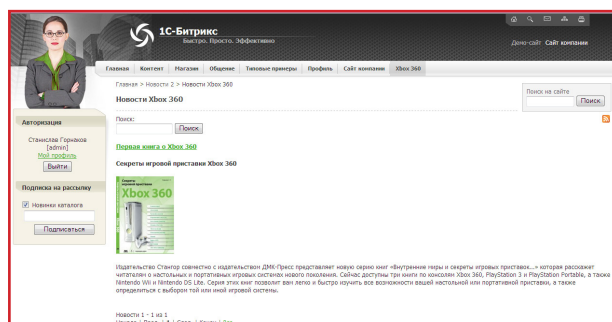


Размещаем компонент новостей на странице сайта

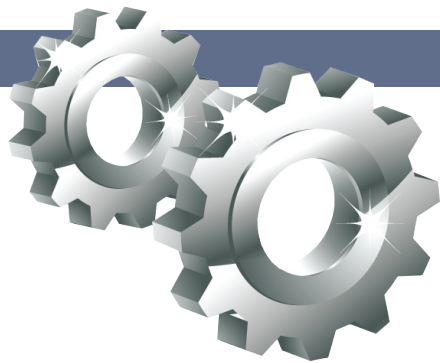
9. Для того чтобы пользователи вашего сайта могли получить доступ к странице с новостями необходимо добавить страницу в меню. Перейдите в режим редакции сайта, выбрав в административной панели вкладку **Редактирование**. Около линейки меню сайта располагается кнопка с одноименным названием. Нажмите кнопку **Меню => Редактировать пункты меню**. Откроется диалоговое окно **Редактирование меню** с перечислением всех пунктов меню сайта. Само окно выполнено в виде таблицы с несколькими колонками. Чтобы добавить новый пункт меню выберите кнопку **Вставить пункт**. В нижней части списка пунктов меню добавится поле для вставки названия пункта меню и ссылки на страницу (колонки **Название** и **Ссылка**). Нажмите курсором на фразу **Нет значения** в колонке **Название**, и впишите название пункта меню. В колонке **Ссылка** нужно указать путь к странице, относительно корневой папки. У нас это папка news2, поэтому пишем /news2/. Это указание пути будет работать только в том случае, если вы создавали в этой папке индексную страницу (index.php). Если же для страницы было выбрано другое название, то необходимо прописывать полный путь с названием страницы (например, /news2/news2.php). Дополнительно с правой стороны диалогового окна **Редактирование меню** располагается ряд графических элементов. Так зеленые стрелочки позволяют передвинуть пункт в иерархии меню. Значок с папкой откроет диалоговое окно с возможностью выбора или поиска искомого файла/страницы в директории сайта, а красный крестик удаляет выбранный пункт меню. Для сохранения добавленного пункта меню нажмите кнопку **Сохранить**. После этого перейдите на созданную страницу и посмотрите результаты работы новостного компонента и информационного блока системы. ■



Редактирование пунктов меню



Страница с добавленными новостями



# Каталог CMS



## ABO.CMS

Компания: ООО «Программные продукты»  
Русификация: Да  
Стоимость: от 0 – до 27300 руб.  
Сайт системы: [www.abocms.ru](http://www.abocms.ru)



## Joomla

Компания: Joomla.org  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: [www.joomla.org](http://www.joomla.org)



## Dakota CMS

Компания: Dakota CMS  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: <http://dakota-cms.com>



## LiveStreet

Компания: [livestreet.ru](http://livestreet.ru)  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: <http://livestreet.ru>



## CMS «Скульптор»

Компания: МЦДИ «БИНЕК»  
Русификация: Да  
Стоимость: от 196руб./мес.  
Сайт системы: <http://support.binec.ru>



## RadioCMS

Компания: RadioCMS  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: <http://radiocms.ru/>



## Drupal

Компания: Asquia  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: [www.drupal.ru](http://www.drupal.ru)



## PHPShop

Компания: ООО «ПХПШОП»  
Русификация: Да  
Стоимость: от 0 до 16770 руб.  
Сайт системы: [www.phpshop.ru](http://www.phpshop.ru)



## UMI.CMS

Компания: Юмисофт  
Русификация: Да  
Стоимость: 0 - 29990 руб.  
Сайт системы: [www.umi-cms.ru](http://www.umi-cms.ru)



## Social Engine

Компания: Webligo Developments  
Русификация: Да  
Стоимость: от 250\$  
Сайт системы: [www.socialengine.ru](http://www.socialengine.ru)



## InstantCMS

Компания: InstantCMS Team  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: [www.instantcms.ru](http://www.instantcms.ru)



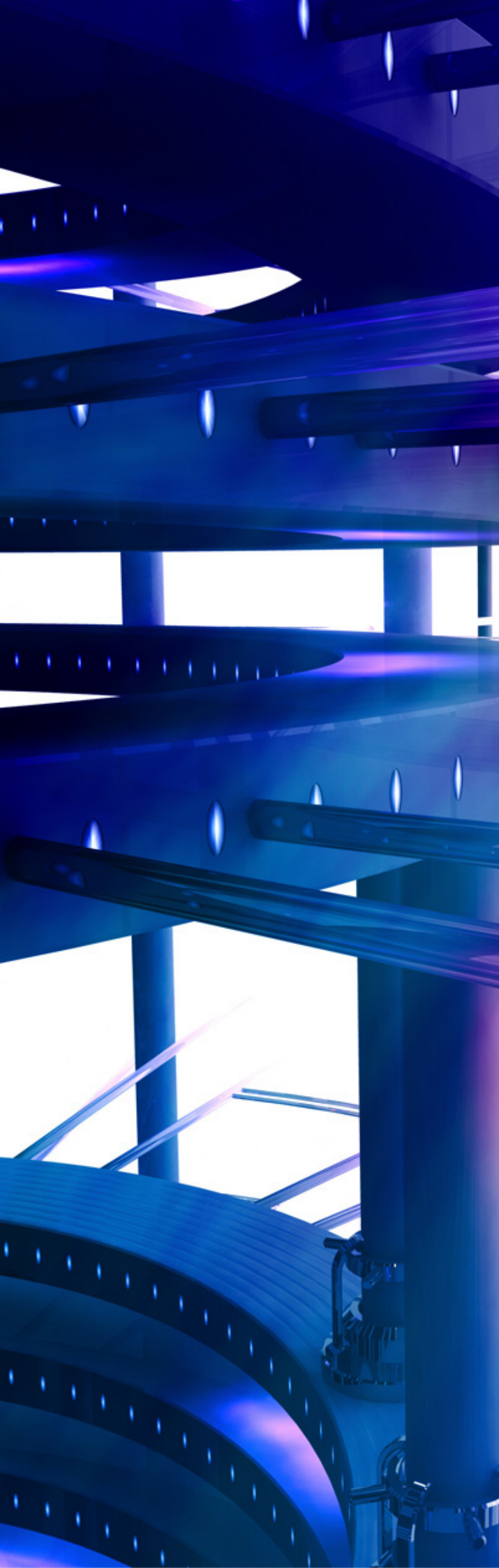
## WordPress

Компания: [WordPress.org](http://WordPress.org)  
Русификация: Да  
Стоимость: Бесплатно  
Сайт системы: [www.wordpress.org](http://www.wordpress.org)

**Добавить CMS в каталог журнала бесплатно**



Публикуемые в журнале компании взяты из каталога сайта издания. Данные о компаниях публикуются по мере их добавления в каталог. Добавить в каталог данные о компании может любая организация. Добавление данных в каталог сайта осуществляется бесплатно на добровольной основе самой организацией. Редакция журнала Веб-Аналитик.ИНФО не занимается добавлением данных в каталог сайта и не несет ответственности за предоставляемую информацию от организаций.



## **ВЕБ-РАЗРАБОТКИ**

**Интервью с  
Сергеем Боровиковым  
[ Руководитель Softline Cloud  
Services  
Компания Softline  
[www.softline.ru](http://www.softline.ru) ]**

**Простое управление проектами  
и коммуникациями с  
подрядчиками для заказчиков  
и владельцев проектов**

**Специализированный софт:  
купить или написать?**

**Требования к html-верстке**

**Пишем OpenID авторизацию**

**Пишем фильтр «плохих» слов**

**Обзор алгоритмов  
кластеризации данных**

**Сказ о том, как Bash и SVG  
спасли от рутины**





# Интервью с Сергеем Боровиковым

**И Руководитель  
Softline Cloud Services  
Компания Softline  
www.softline.ru**

**М**одель «программное обеспечение как услуга» (SaaS) удивительным образом сочетает в себе выгоды для всех участвующих сторон. Для разработчиков это возможность быстро выйти на рынок или охватить новые категории пользователей. Для провайдеров это способ предложить клиентам как хорошо известные, так и новые решения в привлекательной форме. Для конечных пользователей «софт как сервис» — это удобство, простота и экономия. Но что именно получают заказчики, обратившись к сервисам, и как выглядит полученная выгода в материальном выражении?

*Сегодня мало кто сомневается, что программные решения в виде сервисов всерьез конкурируют с традиционным софтом, захватывая зна-*

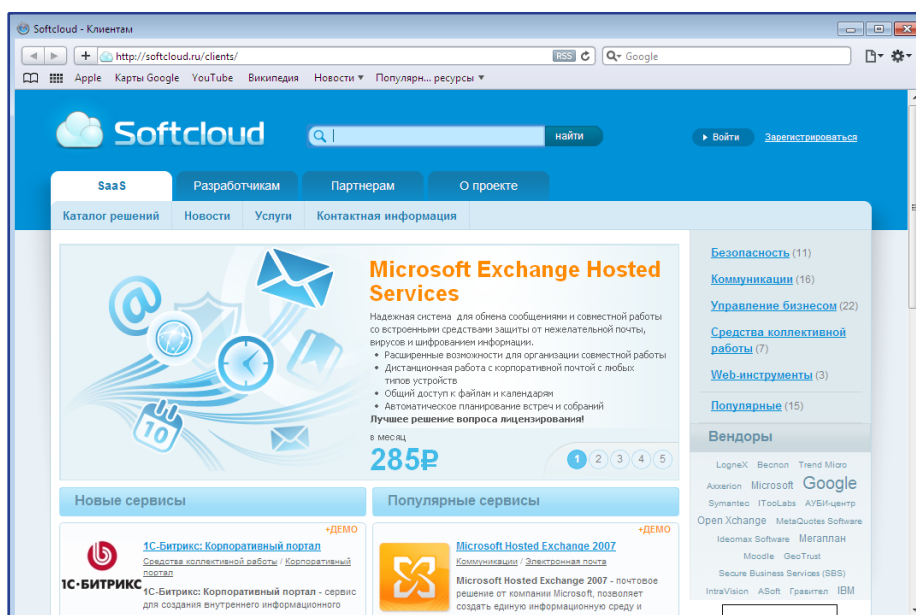
*чительную долю рынка. Какие именно категории решений вы считаете наиболее востребованными и перспективными?*

Я уверен, что очень многие прикладные программы, которые мы привыкли устанавливать на компьютер или сервер с диска, удобнее и выгоднее использовать в формате сервиса. Речь идет о системах хранения и обработки самой разнообразной информации, например о CRM, системах обеспечения безопасности, средствах совместной работы и офисных приложениях, программах для управления проектами и об интернет магазинах. Отдельного внимания заслуживают системы электронной почты. Для них модель SaaS — «то, что доктор прописал».

Во-первых, web-почта появилась

давно, и в сознании людей уже сложилось понимание, что электронная почта может находиться где-то далеко в Интернете, это работает и это действительно удобно. Во-вторых, доступ к почтовому ящику практически всем действительно необходим из разных мест — из офиса, из дома, в командировке или отпуске. В-третьих, почта нужна компаниям во всех отраслях, а это означает огромный рынок, на котором мы можем предложить клиентам множество надежных, проверенных, богатых функциональными возможностями сервисов, рассчитанных на разных заказчиков с разными потребностями. И обходятся эти сервисы гораздо дешевле, чем те же решения в традиционном виде.

*Речь идет о SaaS версиях существующих сервисов?*



*ваших ранее почтовых серверов или о принципиально новых продуктах, впервые появившихся в виде сервисов?*

И то, и другое. Мы предлагаем множество почтовых сервисов для организации корпоративной электронной почты. Некоторые из них не имеют аналогов в мире традиционного «софта», некоторые являются наследниками уже зарекомендовавших себя программных продуктов. К последним относится Microsoft Hosted Exchange – прямой потомок почтового сервера от Microsoft, унаследовавший его возможности, как в области организации электронной почты, так и в средствах коллективной работы.

Microsoft Hosted Exchange – само по себе решение высочайшего уровня, однако мы предлагаем способ сделать его еще лучше – использовать в связке с сервисом фильтрации почтового трафика компании Symantec. Владение этими решениями в традиционном виде обходится очень дорого.

Высокими оказываются не только первоначальные вложения в лицензии и оборудование, но и последующие эксплуатационные расходы: нужно где-то размещать физические серверы, нужно платить зарплату администратору, который их обслуживает. Надежное энергоснабжение, охлаждение, резервное копирование – это далеко не полный список проблем, которые приходится решать.

И речь идет не только о прямых финансовых расходах, но и о головной боли для собственника бизнеса и IT персонала. SaaS действительно эффективно решает эти проблемы. Вернее, в случае с SaaS проблемы просто не возникают. Вам нужно десять почтовых ящиков, надежно защищенных от

вирусов и спама, и вы получаете именно десять таких ящиков, а не сложные и малопонятные системы, которые на самом деле и не нужны. Если через месяц нужно будет сто ящиков, вы немедленно их получите. А абсолютно все проблемы с техническим обеспечением ложатся на плечи сервис-провайдера.

Если говорить о системе фильтрации электронной почты, то стоит упомянуть еще один важный фактор: известно, что порядка 90% электронной почты составляет спам. Если фильтрация спама происходит на вашем собственном сервере, вы тратите время и трафик на их скачивание. Если же фильтрация происходит удаленно, в «облаках», за скачивание спама платить не придется. Разумеется, возможность заглянуть в корзину с удаленными письмами и посмотреть, не отфильтровалось ли что-то нужное, у вас остается. Впрочем, она требуется все реже: решения Symantec отличаются высочайшей точностью распознавания спама – выше 99,9%.

Еще одна важная возможность решения Symantec: проверка подвергаются не только сами сообщения и вложения, но и ссылки, содержащиеся в письме (потенциально опасные ссылки помечаются).

*Как выглядит внедрение Microsoft Hosted Exchange с точки зрения конечного пользователя?*

Если под «конечным пользователем» понимать сотрудника, который сидит за своим рабочим местом и пользуется почтой, то с его точки зрения не меняется ничего. Он ничего не заметит, продолжив работу все в том же Outlook с его папками и календарями.

Для компании процесс внедрения

проходит тоже без больших потрясений. Решив «вынести почту в облака», вы подписываетесь на сервис Exchange. Что происходит дальше? Вариант первый: вы решили завести новую почту с чистого листа. Тогда ваши почтовые ящики заработают в тот же день. Вы получите несложные инструкции по настройке Outlook, по использованию web интерфейса, и по настройке мобильных клиентов. Вот, собственно, и все – можете пользоваться. Вариант второй: у вас уже есть почтовый сервер Microsoft Exchange, неважно какой версии, и неважно, лицензионный или нет. Забрать с него имеющиеся почтовые ящики и перетащить в сервис не составит большого труда. Может быть, это займет один день. Вариант третий: у вас уже есть почтовый сервер, но не Exchange; вариант четвертый: ваша почта находится на некоем бесплатном почтовом web-сервисе. Это наиболее сложные случаи, тут может понадобиться определенный объем ручной работы, но мы умеем это делать, и я не думаю, что процесс миграции затянется дольше, чем на 2-3 дня. И, наконец, приятное дополнение. Подписчики Microsoft Hosted Exchange получают также лицензионный Microsoft Outlook 2007 – важное преимущество для тех, у кого по каким-то причинам нет пакета Microsoft Office.

*Можно ли использовать другие почтовые клиенты?*

Можно пользоваться любыми почтовыми клиентами, понимающими протокол POP3 или IMAP. Правда, при этом вы теряете возможность пользоваться всеми функциями, которые доступны в связке Exchange\_Outlook при работе через протокол MAPI, поддерживаемый ▶

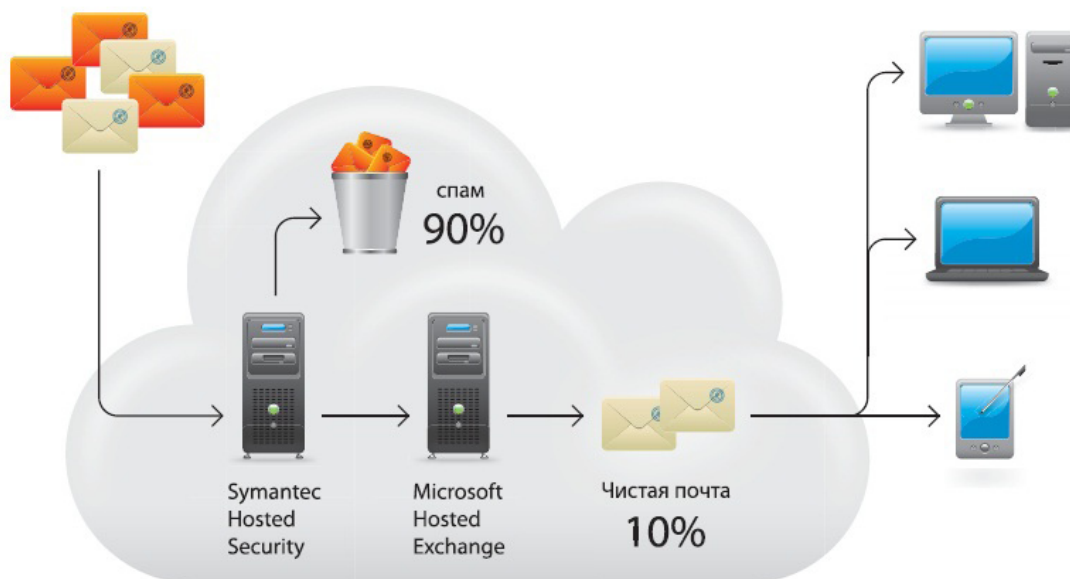


Рис. 1. Схема работы решения Microsoft Hosted Exchange + Symantec Hosted Security

Outlook и почтовым клиентом Windows Mobile.

*Какая ответственность сер-вис-провайдера прописана в соглашении, которое предлагает заказчикам Softline Cloud Services?*

Мы подписываем с заказчиками два соглашения, которые определяют нашу ответственность. Первое соглашение — об уровне обслуживания, в котором прописаны наши обязанности: доступность почты, резервное копирование, техническая поддержка. Второе соглашение — о конфиденциальности, т. е. о неразглашении информации клиента.

*Вы можете рассказать о переходе на SaaS и полученных преимуществах на примерах?*

Очень показательным, можно сказать хрестоматийным, стал недавний переход одного из наших клиентов на то решение, о котором я говорил, на связку Microsoft Hosted Exchange и Symantec Hosted Security. К сожалению, заказчик не разрешил его называть, поэтому могу лишь сообщить, что это дистрибьюторская компания среднего размера. В штате компании один системный администратор, который обслуживает все информационные системы. Несмотря на его высокую загрузку, руководство не планирует расширять технический штат, справедливо полагая, что компания должна заниматься основным бизнесом, а не решением технических вопросов. Перед заказчиком стояла следующая проблема: в сети был установлен нелегальный Microsoft Exchange 2007, в котором использовалось 32 почтовых ящика. Ру-

ководство компании не хотело больше заниматься операциями по его эксплуатации, нести связанные с этим расходы и занимать голову этими проблемами.

Для того чтобы показать эффективность предлагаемого нами решения, мы сделали достаточно точный расчет, который показал, что стартовые вложения для внедрения обычного почтового сервера Exchange составят свыше \$11 тысяч, а его последующая эксплуатация почти \$4 тысяч в год. Те же 32 почтовых ящика в рамках нашего сервиса будут обходиться примерно в \$3600 в год. Экономия за первые три года составляет около 42%. Компания, о которой мы говорим, помимо центрального офиса в Москве имеет филиалы в Санкт-Петербурге и Новосибирске. И почта в формате сервиса подходит для такого территориально распределенного заказчика гораздо лучше, чем любые другие решения. Ведь теперь им не нужно думать о том, как обеспечить доступ к почтовому серверу из разных офисов. Достаточно настроить почтовый клиент на каждом рабочем месте, причем настройки будут одинаковыми.

Еще один плюс: в удаленном офисе подключение к Интернету может быть не таким быстрым и не таким дешевым, как в столичных городах. И экономия на трафике, которую дает удаленная фильтрация почты (помните: спам составляет до 90% почты), может стать весьма существенной. Учитывая приведенные нами расчеты и другие преимущества предложенного решения, заказчик принял решение отказаться от прежней системы в пользу связки Microsoft Hosted Exchange и Symantec Hosted Security. Миграция существующих почтовых ящиков заняла два дня, а на третий день с утра корпоративная почта заработала уже как сервис. Однако с точки зрения сотрудников этот день не отличался от предыдущих — их почтовые клиенты работают точно так-

же, как и раньше, и менять что-то в повседневной работе нет необходимости. Впрочем, одно изменение произошло: перестал приходить спам.

*Какой отклик вы получили от заказчика?*

В ходе предпроектных консультаций мы много общались с генеральным директором. Он очень вменяемый человек, сам в прошлом «айтишник», прекрасно осознающий стоящие перед компанией технические проблемы и знающий цену их решения.

После завершения проекта мне было очень приятно получить от него письмо с благодарностью, в котором он сообщил, что теперь у него стало гораздо больше времени на текущие дела и стратегическое планирование, связанные с основным бизнесом. Ведь именно время обычно является самым ценным ресурсом для руководителя. Откуда взялось лишнее время? Во-первых, имея в распоряжении только одного системного администратора, он вынужден вникать в ситуацию с ИТ, принимать решения. Теперь одной заботой стало меньше. Во-вторых, много времени, до часа ежедневно, раньше уходило на разгребание почты и удаление спама. Теперь спам в почтовый ящик не попадает вовсе. Ситуация, о которой я рассказал, раскрывает характерные для всех компаний проблемы и решения. Я уверен, что аналогичным образом могут облегчить себе жизнь многие компании любых сфер деятельности. Со своей стороны, мы готовы помочь им выбрать и внедрить SaaS-решения, способные принести бизнесу реальную пользу. Приглашаю всех к взаимовыгодному сотрудничеству! ■

#### Сравнение расходов на инфраструктуру электронной почты и фильтрации спама

| Статья расходов                                                                     | Традиционное решение, \$ | SaaS-решение, \$ |
|-------------------------------------------------------------------------------------|--------------------------|------------------|
| Microsoft Exchange Server                                                           | 2680                     | 0                |
| Microsoft Windows Server                                                            | 732                      | 0                |
| Антивирус/ антиспам                                                                 | 450                      | 0                |
| Покупка серверов                                                                    | 3500                     | 0                |
| Помещение, электропитание, поддержка сервера (в год)                                | 400                      | 0                |
| Системный администратор (в год)                                                     | 3500                     | 0                |
| Услуга безопасной почты Microsoft Hosted Exchange + Symantec Hosted E-mail security | 0                        | 3667             |
| <b>Итого в первый год</b>                                                           | <b>11262</b>             | <b>3667</b>      |
| <b>Итого в каждый последующий год</b>                                               | <b>3900</b>              | <b>3667</b>      |



Автор

Сергей Мишин

# Простое управление проектами и коммуникациями с подрядчиками для заказчиков и владельцев проектов

Все большее значение в работе современных компаний приобретает проектная деятельность. Бывают маленькие проекты, например печать не большого тиража брошюр, бывают большие (открытие сети региональных представительств), но все проекты объединяет общая черта – для их выполнения нужно взаимодействие участников и коммуникации.

Сегодня на рынке имеется большое предложение программ и сервисов для управления проектами, но почти все они ориентированы на внутреннее использование или на компаний-разработчиков. Но подавляющее большинство проектов делается компаниями для себя с привлечением к их реализации различных подрядчиков (примеры те же – от брошюр до новых направлений бизнеса)

В этой статье мы расскажем о сервисе qTrack, который ориентирован на использование в первую очередь заказчиком, владельцем проекта (портфеля проектов), а не только исполнителем (подрядчиком).

## Что такое SaaS

Для начала несколько слов о модной сегодня модели распространения, которую использует сервис qTrack. Это так называемый SaaS (Software as a service) – программное обеспечение как услуга. Основной принцип – провайдер услуги берет на себя все обеспечение работоспособности ПО, а от клиента требуется только доступ в интернет. Самый похожий пример – сервисы бесплатной почты, такие как mail.ru или gmail.

Раньше, при использовании классической модели, внедрение и поддержка ПО требовала существенных издержек, начиная от покупки лицензий и постановки их на баланс, до всем известных технических проблем – нужно выделить сервер, установить и настроить ПО, нужен системный администратор, организация создания и хранения бэкапов, установка обновлений и т. д.

*SaaS-модель позволяет забыть об этих проблемах и получить именно то, ради чего используется ПО – новые, более эффективные способы работы*

Также, аренда ПО позволяет легко

начать использовать софт без внедрения по всей компании, а на уровне нескольких сотрудников или отдела. А небольшую стоимость аренды значительно проще согласовать.

## Почему нет приложений для владельцев проектов?

Точнее, нет сервисов для совместной работы команды владельца проекта и привлекаемых сторонних компаний, внештатных сотрудников, фрилансеров.

С точки зрения рынка ситуация довольно проста. С одной стороны есть острый спрос от компаний-разработчиков (выполняющих проекты на заказ), а с другой – еще не полностью сформировавшиеся потребности владельцев проектов (т. е. заказчиков). Получается, что все сервисы делаются под уже имеющийся спрос, потому что понятно, какой должен быть сервис и кому и как его продавать.

Но есть и другая, более жизненная причина – очень многие продукты для управления проектами были созданы для себя теми, кто испытывает наибольшую потребность в автоматизации проектной деятельности – софтверными компаниями. А уже потом они вышли на рынок.

## Зачем нужен сервис для управления коммуникациями?

На первый взгляд может показаться, что владельцу проекта не нужна система автоматизации – у него же есть подрядчики. Но давайте рассмотрим простой пример: создание более-менее крупного web-сайта компании. Времена, когда можно было заказать «сайт под ключ» давно прошли, для получения настоящего бизнес инструмента требуется привлечение разнопрофильных специалистов, которые работают в разных компаниях.

Подрядчики могут быть такие:

- Дизайн-студия (графика, креатив)
- Софтверная компания (программирование)
- Копирайтер (тексты)
- Хостинг-провайдер (размещение сер-

Разработчик: QSOFT ([www.qsoft.ru](http://www.qsoft.ru))

Сайт программы: [www.qtrack.ru](http://www.qtrack.ru)

Языки: русский/английский

Стоимость: бесплатная версия/платная версия от 350 до 3000 рублей в месяц

вера)

- SEO-оптимизатор (продвижение)

Все они должны взаимодействовать не только друг с другом, но и с командой проекта со стороны заказчика – дизайн нужно согласовать с руководством, технические вопросы с IT-отделом и т. д. и т. п.

*От эффективности взаимодействия всех участников проекта зависит и качество, и сроки, и бюджет проекта*

А теперь представьте, что все взаимодействие по проекту осуществляется в личной почте менеджера, а он заболел или увольняется. Это может дорого стоить.

## В чем особенность qTrack?

Разработчики сервиса qTrack не пошли привычным путем и сделали сервис управления проектом для заказчика.

*Ключевая особенность qTrack заключается в возможности работать не только со своей командой, но и с внешними участниками проекта, руководить проектом через управление коммуникациями*

qTrack это не классический трекер задач, а некий гибрид из электронной почты и системы управления проектами.

Функциональные возможности сервиса неразрывно связаны с базовыми принципами правильного управления проектами (разработчик qTrack компания QSOFT, имеет большой опыт управления сложными IT-проектами), соблюдая которые можно избежать типовых ошибок не очень опытных менеджеров. qTrack позволяет: ►

✓ Ставить задачу без предварительной регистрации пользователей – постановка задачи очень похожа на написание обычного электронного письма. При этом ответственный получит письмо с заданием, и ссылкой на вход в сервис.

✓ Работать в привычной электронной почте, что дает возможность взаимодействовать со всеми участниками проекта, не заставляя их осваивать новый софт.

✓ Собирать вместе все знания и историю обсуждения по задаче. Поэтому задача в qTrack представляет собой ленту обсуждения. Это позволяет получить всегда актуальные знания о проекте

✓ Создавать единое пространство для Заказчиков, Исполнителей и Партнеров.

✓ Легко начать работу – дружественный интерфейс сервиса похож на почтовые сайты, а система встроенных подсказок поможет быстрее разобраться с новыми возможностями.

Из дополнительных возможностей стоит отметить весьма мощный список дел (to-do list), который поддерживает иерархию задач и драг-н-дроп (перетаскивание задач мышью). Примечательно, что в списке дел могут быть как простые текстовые задачи (для себя), так и тикеты с обсуждением, что безусловно удобно для менеджера проекта.

## Приватные сообщения

Основной объект в qTrack это тикет – задача, обсуждение и письма в одной сущности (тикет похож на топик в форуме или блог). Работа с тикетами намного удобнее, чем с письмами в почте. Например, есть индикатор, показывающий на чьей стороне вопрос, коллективный доступ к тикету, тэги, гибкая фильтрация.

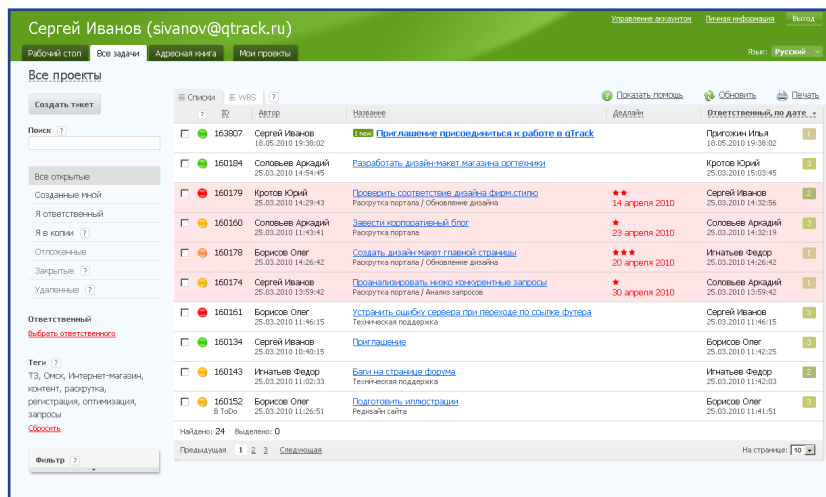
Но наиболее интересная возможность – система скрытых (приватных) сообщений внутри тикета.

*В рамках одного тикета разные группы пользователей могут вести свои ветки частных обсуждений (скрытых от других участников)*

Например, если в тикете идет работа над дизайном страницы, исполнители могут скрыто обсуждать процесс создания, а заказчик со своими коллегами обсуждать, что им нравится, а что нет, после чего менеджер сможет написать открытое сообщение с окончательным результатом обсуждения.

Но это еще не все. Приватные группы не нужно настраивать заранее, они формируются «на лету» исходя из того, какие сообщения были написаны.

Для пользователя логика проста – если он пишет кому-то приватное сообщение, он тем самым добавляет получателя в свою приватную группу (только в рамках текущего тикета). Естественно, получателем сообщения может быть любой человек имеющий e-mail, а



не только пользователь qTrack.

В случае использования проектов и команд все еще интереснее – члены одной команды видят приватные сообщения друг-друга, а состав команд настраиваются индивидуально под каждый проект.

*Все это позволяет не дублировать информацию и не тратить время на «двойной учет». Постановка задачи, история ее изменения и частные мнения участников собраны в одном месте*

## Стоимость

Приятно, что есть полноценная полностью бесплатная версия (доступен только 1 проект).

Стоимость зависит только от количества проектов и неограничивает количество пользователей – от 350 до 3000р. в месяц.

### Достоинства:

+ SaaS-модель – ничего не надо устанавливать, настраивать и поддерживать

+ Работа с внешними партнерами: заказчиками и подрядчиками через привычный e-mail

+ Простой интуитивно понятный интерфейс

+ Начало работы без обучения

+ Постановка задач без предварительной регистрации пользователей

+ Полная интеграция с электронной почтой: отправка письма с заданием, прием ответов в систему, создание ящиков на сервисе и правил к ним

+ Система приватных сообщений

+ Всегда актуальные знания по проекту

+ Мощный список дел (to-do list) с поддержкой иерархии и драг-н-дропа

+ Есть бесплатная редакция

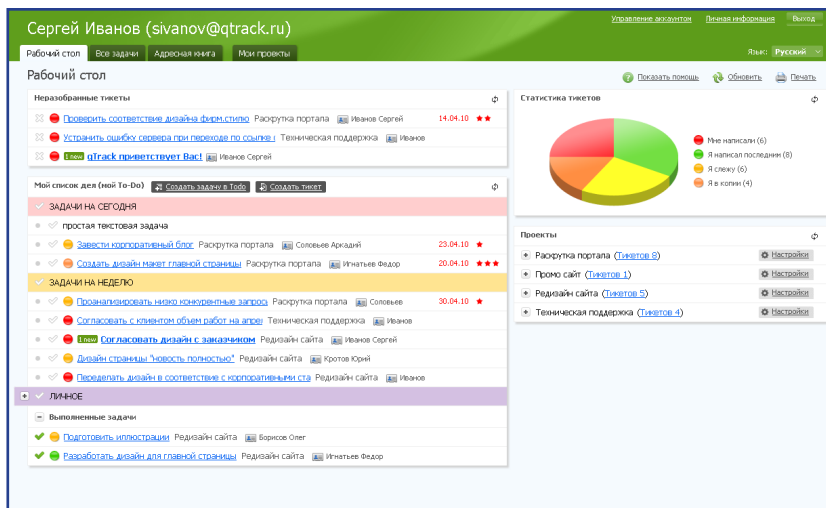
### Недостатки:

– Сервис qTrack дает мощные возможности управления коммуникациями, но ему не хватает средств планирования времени, таких как календарь и диаграмма Ганта.

– Также замечен эргономический просчет – при переходе по ссылке из письма, тикет открывается в отдельном окне, и для перехода к основному интерфейсу нужно нажать неприметную кнопку вверху слева.

\*\*\*

Как говорится, лучше один раз увидеть, чем сто раз услышать – у qTrack нет демо-версии, потому что для регистрации достаточно ввести e-mail и пароль, и посмотреть все «вживую» или даже сразу начать работать. ■



# Специализированный софт:

купить  
или написать?

Приветствую читателей журнала Веб-Аналитик.ИНФО! Думаю, многие сталкивались с проблемой выбора программного обеспечения для каких-либо специфических задач. В отличие от засилья всевозможных редакторов или клиентов для социальных сетей (Facebook, ВКонтакте и т. п.), подобрать узкоспециализированную программу крайне тяжело. Такой софт – продукт нишевый, соответственно производители такого софта малочисленны и интерес у сообщества к подобным программам не велик (в сети нет ни руководств, ни обзоров, ни другой полезной информации) – приходится двигаться на ощупь.

Вот тут и наступают муки выбора – взять готового «кота в мешке» или же взвалить на себя бремя разработки софта «для себя любимого».

Мне не повезло пройти оба пути.

Начинал я свою рабочую деятельность в компании – купи-продай дистрибьюторе. Мы покупали большими партиями бытовую химию и косметику у разных производителей и продавали с доставкой великому множеству крупных и мелких магазинов. Кроме склада и десятка автомобилей в компании трудилась куча-мала торговых представителей, которые ездили по клиентам и собирали заказы. Именно эту часть работы я и хотел оптимизировать.

Дело в том, что каждый торговый агент с утра до ночи собирал заказы (просто записывал на бумажку), а вечером возвращался в офис и надиктовывал эти заявки операторам, которые вбивали их в базу данных (1С). Все это занимало достаточно много времени – приходилось собирать и загружать в автомобили заказанные товары всю ночь.

Я начал искать решение. Сразу стало понятно, что в этой схеме лишние операторы. Если бы торговые представители (или клиенты) напрямую вносили заказы в базу данных, минуя бумажные носители, да еще в реальном времени, то было бы просто супер. Такие решения на рынке уже были – это специальные программы, работающие на КПК с возможностью обмена через GPRS/Wi-Fi. Из не очень большого числа предложений я выбрал решение от компании, которая была на тот момент лидером отрасли. Тут же купил пару

КПК и начал экспериментировать. Все заработало далеко не «с пол пинка», но все же заказы начали приходить напрямую в базу данных в течение дня, и от операторов, и от ночной смены на складе в итоге мы отказались, без потери скорости доставки товаров клиентам.

Омрачало только то, что сам софт был не очень удобен. А все из-за того, что разработчики с каждым релизом вносили все больше и больше функций, которые часто были плохо продуманы, а иногда и откровенно лишние. Программа начала напоминать какого-то неповоротливого монстра. Работать с ней – это как использовать Photoshop для просмотра фотографий...

Года два я терпел, точнее даже не я, а торговые представители, я же периодически их подлавливал, когда они вечером сидели и перебивали заявки из старых добрых бумажек в новые модные КПК. Тогда я решился на написание заказного решения под свои нужды. Нашел компанию-разработчика, составили ТЗ и запустили в «производство».

**Что я хотел получить в этом новом решении?**

- Во-первых мне не понравилось, как я внедрял текущий продукт в свою учетную систему. Тут были сложности с тем, что структура данных, которыми обменивались КПК-клиенты с БД (1С) была плохо продумана, из-

Автор

Алексей Плутенко  
Руководитель проекта «IMTA»  
Компания IMESC  
Украина, г. Киев  
www.imesc.com



лишне сложна и запутана. Кроме этого для организации обмена требовалось «поднять» кучу дополнительных подсистем: отдельный сервер с белым IP; службу, которая принимает данные по неопisanному протоколу; службу Message Queue; парочку плагинов (внешних компонент) для 1С и при этом фактически обмен происходил с помощью простых файлов, «расшаренных» по сети. То есть мне была необходима программа, максимально простая в интеграции с другими системами, без лишних звеньев и сторонних компонент.

- Во-вторых, интерфейс, о проблемах с которым я уже писал выше. Новая программа однозначно должна быть удобна в обращении.
- В-третьих, программа должна содержать только необходимый функционал, без излишеств, которые никому не нужны, а только усложняют работу.
- Ну и конечно была масса мелочей, которые хотелось сделать лучше. ►



## Небольшое сравнение ключевых технических моментов:

|                     | Стороннее решение                                                                                                                                                                                                                                                                                   | Собственное решение                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Процесс обмена с БД | Связь напрямую по TCP/IP между КПК и «офисным сервером». Специальный сервис слушает TCP-порт, получает и раскладывает данные по каталогам файловой системы, и «дергает» 1С через Message Queue и специальную dll-ку, а 1С в свою очередь уже из файлов загружает данные с помощью еще одной dll-ки. | КПК оставляет данные на ftp-сервере, а 1С забирает данные с этого сервера с помощью всего лишь одного COM-объекта.                                                                                                                                                                                                                                                                                      |
| Юзабилити           | В виду избыточного функционала, интерфейс перегружен всякого рода галочками, списками и ненужными подтверждениями. Наиболее часто востребованные действия делаются в несколько касаний.                                                                                                             | Упрощен доступ к основным функциям (создание заявки, выбор клиента, навигация по списку товаров). Значительно удобнее ввод количества (самый «долгий» процесс в работе) – для ввода можно использовать калькулятор, хардварную клавиатуру или джойстик (мега удобно: качнул направо – добавил ящик/штучку, качнул налево – убавил...), или же можно просто «нащелкать» количество пальцем или стилусом. |

Через определенное время была готова первая бета-версия и я начал тестирование. Несмотря на опасения, что отлов «багов» затянется, я в скором времени получил релиз программы и быстро перевел все свое хозяйство на новый софт тут же ощутив преимущества – вся система практически не требовала какой-либо поддержки с моей стороны (в отличие от прошлого решения, где с завидной периодичностью мне приходилось что-то где-то шаманить). И самое главное – торговым агентам стало гораздо удобнее работать. Бумажки исчезли полностью,

а торговые агенты стали похожи на мобильных геймеров – работают только на КПК.

Спустя некоторое время ко мне обратилась другая компания с просьбой внедрить подобную технологию (кто-то из руководства этой компании увидел моих торговых агентов с КПК «в полях» и очень заинтересовался). Я попробовал, и этот опыт получился настолько удачным, что я продолжил развивать продукт вместе с разработчиком.

Прошу любить и жаловать:

<http://www.imta.com.ua>

## В качестве вывода:

Не стоит бояться изобретать велосипед. Поверьте, никто лучше вас самих не знает ваших потребностей, а вера в «умного дядечку», который все придумает за вас, вряд ли принесет необходимый результат. А потратив немного собственных усилий можно получить не только рабочее решение, но и некоторые другие «плюшки». С другой же стороны, не стоит забывать, что разработка собственного продукта не всегда рентабельна и требует много внимания и усилий по поддержке и развитию – как поступить в каждом конкретном случае решать только вам. ■



**IMTA** Автоматизация дистрибуции и мерчандайзинга

помогает **продавать больше**

**Решение IMTA**

Рано или поздно любая компания-дистрибутор сталкивается с необходимостью автоматизировать свою деятельность и обращает внимание на множество программ для автоматизации, представленных на рынке.

Но когда речь заходит об автоматизации дистрибуции, правильнее говорить не об отдельной программе, а о системе или даже комплексном решении. Дистрибуторам проще и выгодней покупать не софт, а законченное решение, внедренное в собственную учетную систему.

Сегодня решение IMTA помогает вести бизнес многим украинским компаниям, работающим в сфере дистрибуции товаров FMCG.

**Одной из самых привлекательных черт решения IMTA является легкость его внедрения и эксплуатации.**

Всего за несколько часов мы способны установить и наладить систему (в случае если вы используете стандартную учетную систему на базе 1С Предприятие. Торговля и Склад). В отличие от других решений IMTA не требует многодневной отладки, настройки и тестирования. IMTA начинает работать и приносит деньги сразу после внедрения.

IMTA не требует наличия в вашем штате специалиста, в обязанности которого входила бы техническая поддержка системы.

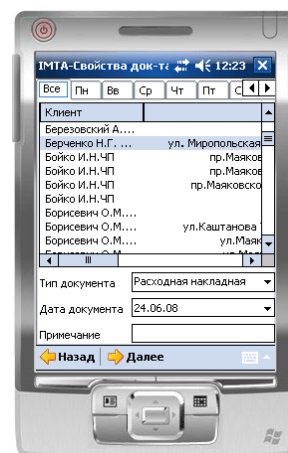
Так же быстро и просто система модифицируется и дополняется новым программными функциями, когда вам потребуется углубленный сбор и анализ маркетинговой информации, дополнительная отчетность торгового персонала и прочее.

**Особое внимание в системе IMTA уделено надежности.**

«Очень понравилась мобильная часть программы - ничего лишнего и все под рукой. Фактически торговый представитель вносит заявку в КПК быстрее, чем оператор на стационарном ПК.»

Буров В.В.,  
Директор  
ООО «НПФ Биопром-Б»,  
Днепропетровск.

Все готовы



# IMTA



## Автор

Виталий Степаненко  
Рекламное агентство «R.A.M.»  
Основатель веб-студии  
Perevaga.com  
www.nayjest.ru

# Требования к html-верстке

**В**ерстка – относительно независимый этап веб-разработки и, к примеру, в маленьких веб-студиях зачастую это первый кандидат на аутсорсинг в условиях ограниченных трудовых ресурсов. Так сложилось, что мне часто приходилось отдавать эту работу субподрядчикам и несмотря на предполагаемую однозначность результата, иногда верстальщики меня очень удивляли. Причем чаще в негативном смысле.

Чтобы сэкономить трудовые ресурсы штатных верстальщиков, недостаточно просто переложить эту работу на плечи первого приглянувшегося фрилансера. Все намного проще, если вы постоянно отдаете работу на аутсорсинг одним и тем же исполнителям – в процессе длительного сотрудничества всегда складывается какой-то негласный свод стандартов и требований, выполнение которых входит в привычку. Но если вы работаете с человеком впервые, то самое хорошее портфолио и рекомендации не гарантируют получения нужного результата и более того, даже не предполагают, что исполнитель вообще вас правильно поймет. Поэтому нужны детальные технические задания по верстке.

И этот момент игнорируется. Часто это происходит из-за предположения, что трудозатраты на написание детального ТЗ в сумме со стоимостью аутсорсинга не окупаются сэкономленным временем штатного верстальщика. В конце концов, «верстка – это ведь не так уж сложно» думает рядовой project менеджер. И, как правило, это действительно прокатывает. (Хвала человеческому интеллекту) профессиональные верстальщики в большинстве своем ограничивают буйство экспериментаторского духа и заранее знают, какие технические решения в верстке могут вызвать у заказчика геморрой не столь адский, чтобы забанить верстальщика, но все же исключая радость и восхищение прекрасным html-макетом.

Тем не менее, вероятность «факапов», как показывает практика, не столь мала, чтобы этим можно было пренебречь. А основное заблуждение здесь в том, что детальное ТЗ – это сложно и трудоемко. Какие-то специфические требования к макету в любом случае приходится описывать, а вот общим требованиям и рекомендациям, как правило, не уделяют внимания.

## О, велика моя скорбь!

Я недавно получил макет, который менеджеры отдавали на аутсорсинг и просто не знал, смеяться мне или плакать. Если бы не мне предстояло интегрировать этот макет в систему шаблонов CMS, я бы наверно все-таки смеялся.

Табличная верстка и стили, не вынесенные в CSS файл и стандартный дримвировский скрипт для подсветки кнопок даже не воспринимаются как недостатки после того ада, который я увидел. Все поля ввода были вставлены внутрь тегов label, засунутых в отдельные формы, причем при попытках как-то привести это в человеческий вид, вся верстка разваливалась. CSS классы имели кириллические названия, причем не осмысленные, а вида «.стиль1», «.стиль2». Большинство элементов форм стилизовались каким-то мало известным и до ужаса кривым скриптом на jQuery, некоторые элементы имели одинаковые ID и между ними был JS код, оперирующий как раз этими элементами и получающий их по ID. Верх маразма – это применение в конце документа метода jQuery.css чтобы задать стили, которые ну совсем ничто не мешало просто прописать в CSS файл. А еще хедер сайта вместе с кучей ссылок (шрифтом Tahoma и без сглаживания) был сделан одной картинкой, на которую наложены элементы MAP и AREA. Не буду больше травмировать вашу психику описанием фейлов в этом замечательном макете, т. к. было их там еще бесчисленное количество.

Происшествия такого вот характера побудили меня опубликовать список требований и рекомендаций, которые будут полезны как людям, отдающим макеты на html-верстку, так и собственно верстальщикам. Вы можете переработать эти рекомендации и дополнить ними свое типовое ТЗ на верстку. Многие вещи из перечисленных вполне очевидны, но вы можете извлечь profit из того, что все они собраны в одном месте. Некоторые пункты (например, требования к поддержке браузерами или используемым скриптам) для разных контур специфичны, но я не буду далее писать расплывчатых фраз, чтобы этот список можно было легко скопипастить и заточить под свою специфику работы. ►



## Требования и рекомендации

1. Кроссбраузерность. Сайт должен нормально работать в IE6-8, FF3+, Opera9+, Safari4+, Chrome4+

2. Всегда описывайте цвет фона для body даже если он белый.

3. Если используете CSS хаки, комментируйте что это, и для какого браузера, а лучше используйте `css_browser_selector.js`. Заботьтесь о верстальщиках, которым придется работать с этим макетом после вас.

4. Названия классов и id должны по смыслу соответствовать применению, например: `header`, `menu`, `footer`, `news`.

5. Просьба разделять основные html блоки комментариями. Примерно так:

```
<!--BEGIN FOOTER -->
<!--END FOOTER -->
```

Это нужно для создания из сверстанного html макета шаблонов для CMS, после чего комментарии будут удаляться.

6. Не пренебрегать возможностью использовать GIF/PNG с 8-битным альфаканалом вместо PNG-24, где это возможно.

7. Все что можно сделать без JavaScript, делать без него.

8. Если Javascript кода много – нужно его выносить в отдельный файл. Обработчики событий тоже лучше отделить и объявлять в отдельном файле.

9. Если это еще не оговорено с заказчиком, предварительно оговорить, какие JavaScript библиотеки вы планируете использовать при верстке макета, чтобы потом не оказалось, что при верстке использовался, к примеру, PrototypeJS, а заказчик планирует в обязательном порядке внедрять на сайт jQuery.

10. Для резиновых макетов обязательно должна быть задана минимальная и максимальная ширина.

11. Если в ТЗ не сказано другое, макет обязательно должен помещаться без горизонтальных скроллбаров в развернутое на весь экран окно браузера при горизонтальной составляющей разрешения экрана 1024px, а если позволяет размер макета, то и 800px.

12. В папке с изображениями не должно быть картинок, не использующихся в верстке. Если что-то исключили из верстки или переделали – не

забывайте удалять уже ненужные картинки.

13. Для всех элементов, которые могут содержать текст различной длины, вписывающийся в одну строку (например, для кнопок или заголовков, если в дизайне не предусмотрено, что они могут занимать больше одной строки), обязательно задавайте CSS свойство `white-space: nowrap`.

14. CSS файл должен быть разбит с помощью строк с комментариями на блоки по функциональному назначению, например:

```
/* 1. Сброс CSS */
/* 2. Типовые элементы */
/* 2.1. Заголовки */
/* 2.2. Ссылки */
/* 2.3. Элементы форм */
/* 3. HEADER (Шапка сайта) */
/* 4. FOOTER (Подвал) */
/* 5. SIDEBAR (Справа) */
```

Как именно структурировать стили – вопрос немного холиварный, но главное как-то это все равно делать.

15. Если сдача верстки производится более чем одним этапом (например, верстальщик отправляет страницы по одной, или если ему возвращаются на доработку уже сверстанные страницы) и вы не используете систему контроля версий для верстки, исполнитель должен в обязательном порядке прикрепить файл с описанием изменений в макете примерно такого содержания:

- Добавил новые картинки в папку `img`.
- Картинки `btnHome.jpg` и `btnFeedback.jpg` уже не нужны, можно удалять.
- Изменил html-код в секции файла `anketa.html`.
- Добавил в конец файла `main.css` новые стили (начиная с `.form_row` и ниже).

16. Оговорить, в какой кодировке должен быть html-макет. CSS и JS файлы должны быть в той же кодировке, что и макет, иначе вероятность долгой и утомительной охоты на баги критически возрастает.

17. Если в макете присутствует JS, изменяющий DOM, то внимательно следить, чтобы все корректно работало в Опере, т. к. этот замечательный браузер при нажатии на кнопку **Назад** страницу не перезагружает, а отдает закешированный документ, причем

очень важный момент: документ не просто закешированный, а еще и с учетом всех модификаций DOM, которые были выполнены с помощью JS

18. Не забывайте прописывать `cursor: pointer` для кнопок, сделанных не с помощью `input`. Если вы не знаете, будет на эту кнопку повешен обработчик событий с помощью JS или это будет ссылкой, лучше в любом случае использовать тег `<a>`.

19. Если вы делаете обработку событий при нажатии на ссылки, следите за тем, чтобы обработчики событий возвращали `false`, или же используйте `href='javascript:void(0)'` вместо популярного `href='#'`, чтобы страница не скроллилась вверх.

20. Верстайте формы правильно: метки полей должны находиться в тегах `label`, имеющих правильно заполненный атрибут `for`. Предусматривайте при верстке форм элементы для вывода ошибок валидации и стили для неправильно заполненных полей. Если это не предусмотрено в ТЗ и дизайне, обязательно обсудите это с заказчиком.

21. Если в макете используются нестандартные шрифты, обязательно оговорите, можно ли элементы с нестандартным шрифтом делать картинками, если нельзя – обсудите, с помощью какой технологии будет реализовано их отображение (`slfr`, `Cufon`, etc.)

22. Если не оговорено обратное для частных случаев, все блоки, высоту которых ничего в дизайне не мешает сделать динамической, должны иметь именно динамическую (т. е. зависящую от содержания) высоту, а иногда, чтобы ничего не могло потенциально поломать дизайн, нужно задавать и минимальную высоту. Если хотите сделать блок фиксированной высоты, то сначала спросите у заказчика.

23. В макетах, где высота страницы зависит от контента (а таких, как правило, большинство), предусматривайте, чтобы футер был прибит к низу браузера при отсутствии/малом количестве контента, если не оговорено обратное.

24. Если макет не проходит 100%-ную html-валидацию, постарайтесь по крайней мере делать так, чтобы использование невалидного кода было оправданно. Не стоит факкапить валидность верстки только потому, что «мне так нравится» или «так получается короче». ■





# Пишем OpenID авторизацию

Автор

Николай Марков  
Программист  
г. Москва

В последнее время всякие социальные сети и вообще сервисы-лидеры интернета по посещаемости и количеству аккаунтов завели очень неплохую, на мой взгляд, привычку – предоставление уникальных OpenID-идентификаторов для пользователей, с помощью которых можно зайти на сторонний сайт. Кроме того, параллельно развивается очень похожая, но все-таки не совсем производная технология OAuth, которая появилась на свет благодаря стараниям создателей небезызвестного Twitter и, цитируя википедию, «позволяет предоставить третьей стороне доступ к защищенным ресурсам пользователя, без необходимости передавать ей (третьей стороне) логин и пароль».

Лично меня такая тенденция очень радует и, более того, я почти уверен, что за подобной технологией будущее. В частности, в будущем обязательно появятся новые мзшапы для агрегирования информации с кучи сайтов, а именно такой «форм-фактор» требует логина на множество сервисов сразу. В частности, хочется вспомнить очень хороший, но несправедливо забытый сервис Yahoo Pipes, который так и не смог покорить сердца и умы просто потому, что его время тогда еще не пришло. Возможно, все еще впереди.

Пять дифирамбы подобным технологиям можно очень долго, но лично меня, например, всегда напрягали сайты, на которых надо с нуля регистрироваться чтобы что-нибудь скачать. Ведь все мы неизменно сталкивались с тем, что когда ищешь где скачать тот или иной материал, он зачастую оказывается на каком-то совершенно левом и непонятном сайте с названием в духе allbooksmusicwarezzz.omg.su, который ко всему прочему еще и регистрацию требует. Дело не в пиратстве, дело в том, что сайтов со всяким бардаком, сделанных на коленке, уйма. А вот человеческая память на логины-пароли ограничена и тут уж ничего не сделаешь. Но приятный момент здесь еще и в том, что многие OpenID-провайдеры кроме информации непосредственно служащей для авторизации, могут по запросу предоставить еще и базовую информацию о пользователе – e-mail, полное имя, предпочтительный язык и т. п. Причем на многих подобных сервисах можно управлять тем, что именно отдавать, а что сохранить в секрете. Например, разве пользователю не будет приятно, когда он, зайдя на очередной сайт, увидит приветливую надпись «Добро пожаловать, Вася!» на чистом русском языке, да еще и профиль уже готов к употреблению, вместе с аватаром, привычками и кличкой нежно любимого кота?

## Делаем дело и работаем работу

Довольно лирики, думаю, кому это надо как разработчику, то он и так все вышенаписанное уже знает, а простым пользователям дальнейший материал вряд ли будет интересен. Поэтому давайте попробуем сделать свою систему авторизации через OpenID (например, для блога) на Python, с преферансом и пианистками.

Для своего блога, который сейчас находится в разработке у меня в папке Projects, я решил вообще отказаться от системы регистрации и авторизации через логин-пароль, а оставить только OpenID. В качестве фреймворка был выбран Pylons, а для прикручивания OpenID к Django-проектам суще-

ствует и развивается проект с простым и понятным названием django-openid. Для Pylons, в общем-то, тоже существует решение под названием AuthKit, однако с ним у меня отношения как-то не очень сложились, а все, что я нашел в сети – это несколько сниппетов, в которых и пришлось разбираться.

Для начала надо установить модуль python-openid, чтобы обеспечить поддержку технологии, а потом создаем контроллер (обработчик запроса по URL, ближайшая ассоциация – джанговский views.py) и начинаем колдовать.

```
$ paster controller auth
```

Сразу оговорюсь, что код рабочий ровно до той степени, которая обеспечивает непосредственно аутентификацию, что делать дальше и как это все оформлять – решать только вам, господа творцы. Начало довольно стандартное:

```
from openid.consumer.consumer import Consumer, SUCCESS, FAILURE, DiscoveryFailure
```

```
from openid.store import filestore
```

```
from openid import sreg
```

```
from datetime import datetime
```

```
from hashlib import md5
```

```
class AuthController(BaseController):
```

```
    def __before__(self):
```

```
        self.openid_session = session.get('openid_session', {}) # проверяем, не существует ли openid-сессии
```

```
    def index(self):
```

```
        return render('/accounts/enter.html')
```

```
    @rest.dispatch_on(POST='»signin_POST») # разделяем GET- и POST-запросы по разным обработчикам для удобства
```

```
    def signin(self):
```

```
        if c.user: # проверяем, не попытался ли уже залогиненный юзер зайти еще раз
```

```
            session['message'] = 'Already signed in.'
```

```
            session.save()
```

```
            redirect(url(action='index')) # и если да, то не пушаем
```

```
            session.clear()
```

```
            return render('/index.html')
```

Теперь подходим к самому интересному:

```
    def signin_POST(self):
```

```
        problem_msg = 'A problem occurred communicating to your OpenID server. Please try again.' ▶
```

```
g.openid_store = filestore.FileOpenIDStore('.') # создаем временное хранилище для хранения OpenID-данных, g здесь-массив глобальных переменных Pylons
```

```
self.consumer = Consumer(self.openid_session, g.openid_store) # ага, вот и наш клиент
```

```
openid = request.params.get('openid', None) # достаем из запроса строку с OpenID - идентификатором
```

А вот тут немного магии. SReg – это то самое расширение, которое позволяет нам запросить у сервера дополнительные данные о пользователе. Поля, значение которых хотелось бы узнать, перечисляем в списке `optional`, а дополнительные данные потом по желанию всегда можно запросить у пользователя. Если же какая-то дополнительная информация требуется прямо «кровь из носу», то можно запросить ее в `required`, но если сервер ее не отдаст – будет ошибка.

```
sreg_request = sreg.SRegRequest(
    #required=['email'],
    optional=['fullname', 'timezone',
              'language', 'email', 'nickname']
)
```

```
if openid is None:
    session['message'] = problem_msg
    session.save()
    return render('/index.html')
```

Здесь я позволил себе схалтурить и написал этот код только для того, чтобы объяснить разницу между простым OpenID и кросс-логином с гугловского аккаунта. Дело в том, что гугл не представляет пользователям OpenID-идентификатора вида `https://vasya_pupkin.google.com`. URL идентификации у всех пользователей гугла выглядит абсолютно одинаково – `https://www.google.com/accounts/o8/id`. Любопытно то, что при запросе на этот URL, гугл отдает готовый XRDS (XML-подобного вида документ, возвращаемый сервером по стандарту OpenID 2.0), который уже содержит все необходимое для авторизации, а вам как пользователю присваивается уникальный ID, который и является, по сути, идентификатором OpenID.

```
if openid == 'google':
    openid = 'https://www.google.com/accounts/o8/id'
```

```
try:
    authrequest = self.consumer.begin(openid)
    # панеслася
```

```
except DiscoveryFailure, e: # а вдруг ошибка в адресе или такой провайдер существует только в вашем воображении?
```

```
session['message'] = problem_msg
session.save()
return redirect(url(controller='auth',
action='signin'))
```

```
authrequest.addExtension(sreg_request) # подключаем SReg, дабы извлечь требуемые поля для профиля
```

```
redirecturl = authrequest.redirectURL(h.url_for('/', qualified=True),
```

```
return_to=h.url_for(action='verified',
qualified=True),
```

```
immediate=False
```

```
) # после всего, что у нас было с сервером, надо как-то жить дальше
```

```
session['openid_session'] = self.openid_session
```

```
session.save()
```

```
return redirect(url(redirecturl))
```

Ну, теперь можно поговорить с сервером, останемся ли мы друзьями.

```
def verified(self):
```

```
    problem_msg = 'A problem occurred communicating to your OpenID server. Please try again.'
```

```
    self.consumer = Consumer(self.openid_session, g.openid_store)
```

```
    info=self.consumer.complete(request.params,
(h.url_for(controller='auth',
```

```
action='verified',
```

```
qualified=True)))
```

```
    if info.status == SUCCESS: # все пучком
```

```
        sreg_response = sreg.SRegResponse.fromSuccessResponse(info) # извлекаем затребованные в SReg поля
```

```
        user = User(by_openid=info.identity_url)
        # ищем юзера по идентификатору в базе
```

```
        if not user.exist: # а вот тут можно делать что угодно. Например, внести юзера в базу
```

```
            newuser = User()
```

```
            try:
```

```
                email = sreg_response.get('email', u''),
```

```
            except:
```

```
                email = u''
```

```
            newuser.create(
```

```
                openid = unicode(info.identity_url),
```

```
                email = email,
```

```
                password = unicode(md5(info.identity_url).hexdigest()),
```

```
                ip = request.environ['REMOTE_ADDR']
```

```
            )
```

```
        session.clear() # мутим сессию
```

```
        session['openid'] = info.identity_url
```

```
        session.save()
```

```
        if 'redirected_from' in session:
```

```
            red_url = session['redirected_from']
```

```
            del(session['redirected_from'])
```

```
            session.save()
```

```
            return redirect(url(red_url))
```

```
        return redirect(url(controller='auth',
action='index'))
```

```
    else: # факир был пьян
```

```
        session['message'] = problem_msg
```

```
        session.save()
```

```
        return redirect(url(action='signin'))
```

Вот, собственно, и все. Остается добавить, что данный код не работает с OpenID от Yahoo. А что делать с полученными данными – помещать в куки, продолжать регистрацию и просить у пользователя дополнительную информацию – решать вам. ■

Автор

Андрей "d1" Перминов  
Администратор

# Пишем фильтр «плохих» слов

Многие из вас наверняка не раз бывали в публичных многопользовательских чатах. Будь то форум, вебчат или чат-сервер в локалке провайдера – идеального порядка можно достичь лишь при малом количестве пользователей (не более сотни, на мой взгляд). Когда сообщество разрастается, возникают примерно одни и те же проблемы: мат, спам и флуд – один скучающий индивид каждую минуту постит «Все в кантру!», другой матерится по поводу и без, и все в таком духе.

Идеального мат-фильтра до сих пор не изобретено. Но мы этим заниматься и не будем, рассмотрим реализацию необходимого минимума.

## Мат

Конечно же мы на нем разговариваем, кто же спорит :). С моей точки зрения мат может иметь место для красного словца или выражения пущей экспрессии. Но далеко не все столь легко его воспринимают, и с этим необходимо мириться (или создавать свое сообщество а-ля «матерись сколько влезет»). Особенно неприглядно выглядит поток слаботивированного мата от не слишком умных / воспитанных граждан. Ну, это оставим на их совести.

Русский язык очень богат возможностями слово- и фразообразования, поэтому матерных конструкций придумать можно очень много, порой диву даешься. Так что мы не будем пытаться бороться с этими явлениями. Решение в лоб – фильтр по словарю. Сам словарь, думаю, не составит большого труда собрать. У меня вышло где-то 250 слов.

Но первые же опыты показывают, что это метод не дает положительного результата. Юзеры обучаются обходить его крайне быстро, видя неудачи пары первых погорельцев. Как они это делают? В основном вариантов несколько:

- Замена согласных на парные созвучные им (например «х» на «к»);
- Замена букв на аналогично выглядящие латинские («п» – «p», «и» – «u»);
- Смешение кириллицы и транслита;
- Употребление псевдографики.

На помощь приходят регулярные выражения. Создадим таблицу замены символов с учетом вышесказанного.

|                                       |
|---------------------------------------|
| (                                     |
| 'а' => ['a', 'a', '@'],               |
| 'б' => ['б', '6', 'b'],               |
| 'в' => ['в', 'b', 'v'],               |
| 'г' => ['г', 'r', 'g'],               |
| 'д' => ['д', 'd', 'g'],               |
| 'е' => ['e', 'e'],                    |
| 'ё' => ['ё', 'e', 'e'],               |
| 'ж' => ['ж', 'zh', '*'],              |
| 'з' => ['з', '3', 'z'],               |
| 'и' => ['и', 'u', 'i'],               |
| 'й' => ['й', 'u', 'y', 'i'],          |
| 'к' => ['к', 'k', 'i{', ' {'],        |
| 'л' => ['л', 'l', 'ji'],              |
| 'м' => ['м', 'm'],                    |
| 'н' => ['н', 'h', 'n'],               |
| 'о' => ['o', 'o', '0'],               |
| 'п' => ['п', 'n', 'p'],               |
| 'р' => ['p', 'r', 'p'],               |
| 'с' => ['c', 'c', 's'],               |
| 'т' => ['т', 'm', 't'],               |
| 'у' => ['y', 'y', 'u'],               |
| 'ф' => ['ф', 'f'],                    |
| 'х' => ['x', 'x', 'h', 'к', 'k', ''], |
| 'ц' => ['ц', 'c', 'u'],               |
| 'ч' => ['ч', 'ch'],                   |
| 'ш' => ['ш', 'sh'],                   |
| 'щ' => ['щ', 'sch'],                  |
| 'ъ' => ['ъ', 'b'],                    |
| 'ы' => ['ы', 'bi'],                   |
| 'ь' => ['ь'],                         |
| 'э' => ['э', 'e', 'e'],               |





```
'ю' => ['ю', 'io'],
```

```
'я' => ['я', 'ya'],
```

```
)
```

Теперь нужно сделать из нашего словаря библиотеку регулярных выражений. В принципе их можно конструировать и налету. Принцип анализа прост: на месте буквы может быть любая последовательность из таблицы. Чтобы народ не хитрил вставляя пробелы, точки и т. п. между буквами для обхода фильтра, то их появление между буквами тоже учитываем. Ну и чтобы не получалось ложных срабатываний будем обрабатывать только слово целиком, без учета словообразования сложных конструкций.

Да, последнее конечно несколько снижает эффективность фильтра, но все-таки видоизмененные слова уже не очень-то и мат, а основную массу базовых слов и фраз мы можем отрезать словарем. В принципе если срабатывание фильтра необходимо только для скрывания неприглядных слов (а не для немедленного бан-хаммера), то это условие можно отбросить.

В итоге для всем известного слова из трех букв регулярка будет выглядеть примерно следующим образом:

```
m/(?:^|\W)((?:x|x|h|k|k|){})\W*[yuy]\W*[йуѐ])(?:$|\W)/ig;
```

В общем суть проста: делаем словарь, из него генерируем библиотеку регулярок и применяем их в нашем фильтре.

Недостатки подхода:

- Необходим большой словарь;
- Не учитывается морфология русского языка;
- Возможны ложные срабатывания («Куй железо пока горячо!»)

То есть в общем-то это не панацея, но определенный эффект сдерживания имеет.

## Спам и флуд

Флуд или бессмысленный поток символов, смайлов, предложений может порядком надоедать. Спамом в нашем контексте мы также будем считать разновидность флуда, повторяющиеся фразы вроде «Все ф кантру! Создано на x.x.x» или «Народ, заходим на мою супер мегастраничку link». Хотя эти явления больше характерны для чатов.

Отслеживать все это дело можно чисто механически, без применения какого-либо семантического анализа текста, но есть ли простые методы это сделать?

С флудом, повторяющимися символами и смайлами в

общем-то все понятно: подсматриваем в живом чате самые частые схемы флуда и используя квантификаторы их моделируем. Лишь некоторые примеры:

```
/(\.){15,}/is;
```

```
/(\.){3,}){15,}/is;
```

```
/(\n|r|r\n){13,}/is;
```

Сложнее дело обстоит с повторами фраз. Подход в лоб довольно примитивен: сравниваем n-последних сообщений пользователя и в случае совпадений «срабатываем».

Первейший способ сопротивления – добавляем пробелы в строку, меняем количество знаков препинания (точек, восклицательных знаков). Впрочем, мы ведь можем сделать нормализацию строки, учитывающую такие изменения: например, просто убираем все лишнее. Однако тут нас ждет неудача со второй волной сопротивления: фразу начинают слегка менять, добавлять междометия, вводные слова и т. п. Вот тут уже придутся алгоритмы нечеткого сравнения.

Например, можно вычислять Расстояние Левенштейна. Реализация этого алгоритма не очень сложна и есть на большинстве популярных языков, а в php так вообще присутствует нативно. Фактически, он возвращает разницу строк в количестве правок, необходимых для получения второй строки из первой. Далее дело техники: определяем минимальный размер сообщения, на который стоит реагировать, считаем расстояние и если оно не превышает, например, 20% от длины сообщения, то считаем его повтором! Все просто. Длину сообщения и порог расстояния подбираем эмпирически, чтобы не мешать обычным пользователям.

Этот метод, конечно же, поможет только в противодействии скукающих индивидов, рассылающих все вручную (отсюда и малые правки, и повторяемость фраз). Если же пользователь шлет разнородные сообщения с пугающей скоростью, то, скорее всего это спам-бот и тут нужны другие подходы или действия живого модератора.

## Итог

Итак, мы рассмотрели в общих чертах методы противодействия мату, флуду и спаму. Предложенные методы наиболее актуальны для чат-систем коротких сообщений. Во многом простота подходов обусловлена применением такой в общем-то довольно непростой вещи как регулярные выражения.

Существуют и альтернативные методы, включающие попытки морфологического и семантического анализа фраз. Однако, если следовать идеологии того же AIML, человекоподобное поведение можно получить вполне эмпирически, не встраивая между вводом и выводом «слишком много компьютера». ■

## Автор

Андрей Часовских  
ООО «Брэйн Системс»,  
разработчик  
г. Санкт-Петербург  
www.brainsystems.ru  
Andrey.Chasovskikh@brainsystems.ru

# Обзор алгоритмов кластеризации данных

В современном мире человек имеет дело с огромными объемами информации, которую необходимо как-то перерабатывать и анализировать. В этом помогает интеллектуальный анализ данных (data mining). Одним из инструментов анализа данных является кластерный анализ. С его помощью становится возможным разделять большие объемы информации на более мелкие, и делать репрезентативную выборку из большого множества объектов. Помимо интеллектуального анализа данных, кластеризация также применяется в технологиях поиска и сегментации изображений.

## Понятие кластеризации

Кластеризация (или кластерный анализ) — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы должны оказаться «похожие» объекты, а объекты разных групп должны как можно более различаться. Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости — нормализация значений переменных.
3. Вычисление значений меры сходства между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения результатов и их анализа, возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

## Меры расстояний

Итак, как же определять «похожесть»

объектов? Для начала нужно составить вектор характеристик для каждого объекта — как правило, это набор числовых значений, например, рост-вес человека. Однако существуют также алгоритмы, работающие с качественными (т. н. категориальными) характеристиками.

После того, как мы определили вектор характеристик, можно провести нормализацию, чтобы все компоненты давали одинаковый вклад при расчете «расстояния». В процессе нормализации все значения приводятся к некоторому диапазону, например, [-1, -1] или [0, 1].

Наконец, для каждой пары объектов измеряется «расстояние» между ними — степень похожести. Существует множество метрик, вот лишь основные из них:

### 1. Евклидово расстояние

Наиболее распространенная функция расстояния. Представляет собой геометрическое расстояние в многомерном пространстве:

$$\rho(x, x') = \sqrt{\sum_i^n (x_i - x'_i)^2}$$

### 2. Квадрат евклидова расстояния

Применяется для придания большего веса более отдаленным друг от друга объектам. Это расстояние вычисляется следующим образом:

$$\rho(x, x') = \sum_i^n (x_i - x'_i)^2$$

### 3. Расстояние городских кварталов (манхэттенское расстояние)

Это расстояние является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т. к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния: ►

$$\rho(x, x') = \sum_i^n |x_i - x'_i|$$



#### 4. Расстояние Чебышева

Это расстояние может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

$$\rho(x, x') = \max(|x_i - x'_i|)$$

#### 5. Степенное расстояние

Применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются. Степенное расстояние вычисляется по следующей формуле:

$$\rho(x, x') = r \sqrt[p]{\sum_i^n (x_i - x'_i)^p}$$

где  $r$  и  $p$  – параметры, определяемые пользователем. Параметр  $p$  ответствен за постепенное взвешивание разностей по отдельным координатам, параметр  $r$  ответствен за прогрессивное взвешивание больших расстояний между объектами. Если оба параметра  $r$  и  $p$  равны двум, то это расстояние совпадает с расстоянием Евклида.

Выбор метрики полностью лежит на исследователе, поскольку результаты кластеризации могут существенно отличаться при использовании разных мер.

## Классификация алгоритмов

Для себя я выделил две основные классификации алгоритмов кластеризации.

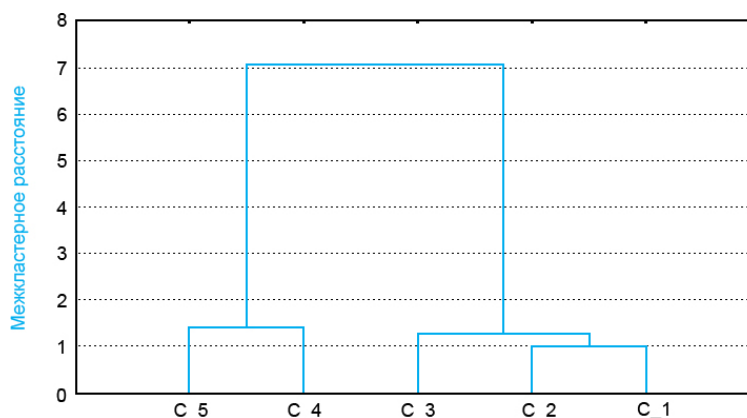
#### 1. Иерархические и плоские

Иерархические алгоритмы (также называемые алгоритмами таксономии) строят не одно разбиение выборки на непересекающиеся кластеры, а систему вложенных разбиений. Таким образом на выходе мы получаем дерево кластеров, корнем которого является вся выборка, а листьями – наиболее мелкие кластеры.

Плоские алгоритмы строят одно разбиение объектов на кластеры.

#### 2. Четкие и нечеткие

Четкие (или непересекающиеся) алгоритмы каждому объекту выборки ставят в соответствие номер кластера, то есть каждый объект принадлежит только одному кластеру. Нечеткие (или пересекающиеся) алгоритмы каждому объекту ставят в соответствие набор вещественных значений, показывающих степень отношения объекта к кластерам. То есть каждый объект относится к каждому кластеру с некоторой вероятностью.



Иерархическая кластеризация

## Объединение кластеров

В случае использования иерархических алгоритмов встает вопрос, как объединять между собой кластеры, как вычислять «расстояния» между ними. Существует несколько метрик:

1. Одиночная связь (single-link или расстояние ближайшего соседа)

В этом методе расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Результирующие кластеры имеют тенденцию объединяться в цепочки.

2. Полная связь (complete-link или расстояние наиболее удаленных соседей)

В этом методе расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т. е. наиболее удаленными соседями). Этот метод обычно работает очень хорошо, когда объекты происходят из отдельных групп. Если же кластеры имеют удлиненную форму или их естественный тип является «цепочечным», то этот метод непригоден.

3. Невзвешенное попарное среднее

В этом методе расстояние между двумя различными кластерами вычисляется как среднее расстояние между всеми парами объектов в них. Метод эффективен, когда объекты формируют различные группы, однако он работает одинаково хорошо и в случаях протяженных («цепочечного» типа) кластеров.

4. Взвешенное попарное среднее

Метод идентичен методу невзвешенного попарного среднего, за исключением того, что при вычислениях размер соответствующих кластеров (т. е. число объектов, содержащихся в них) используется в качестве весового коэффициента. Поэтому данный метод должен быть использован, когда предполагаются неравные размеры кластеров.

5. Невзвешенный центроидный метод

В этом методе расстояние между двумя кластерами определяется как расстояние между их центрами тяжести.

6. Взвешенный центроидный метод (медиана)

Этот метод идентичен предыдущему, за исключением того, что при вычислениях используются веса для учета разницы между размерами кластеров. Поэтому, если имеются или подозреваются значительные отличия в размерах кластеров, этот метод оказывается предпочтительнее предыдущего.

## Обзор алгоритмов

### Алгоритмы иерархической кластеризации

Среди алгоритмов иерархической кластеризации выделяются два основных типа: восходящие и нисходящие алгоритмы. Нисходящие алгоритмы работают по принципу «сверху-вниз»: сначала все объекты помещаются в один кластер, который затем разбивается на все более мелкие кластеры. Более распространены восходящие алгоритмы, которые в начале работы помещают каждый объект в отдельный кластер, а затем объединяют кластеры во все более крупные, пока все объекты выборки не будут содержаться в одном кластере. Таким образом, строится система вложенных разбиений. Результаты таких алгоритмов обычно представляют в виде дерева – дендрограммы. Классический пример такого дерева – классификация животных и растений.

Для вычисления расстояний между кластерами чаще всего используются два расстояния: одиночной связью или полной связью (см. выше обзор мер расстояний между кластерами).

К недостатку иерархических алгоритмов можно отнести систему полных разбиений, которая может являться излишней в контексте решаемой задачи.

### Алгоритмы квадратичной ошибки

Задачу кластеризации можно рассматривать как построение оптимального



разбиения объектов на группы. При этом оптимальность может быть определена как требование минимизации среднеквадратической ошибки разбиения:

$$e^2(X, L) = \sum_{j=1}^K \sum_{i=1}^{n_j} \|x_i^{(j)} - c_j\|^2$$

где  $c_j$  – «центр масс» кластера  $j$  (точка со средними значениями характеристик для данного кластера).

Алгоритмы квадратичной ошибки относятся к типу плоских алгоритмов. Самым распространенным алгоритмом этой категории является метод  $k$ -средних. Этот алгоритм строит заданное число кластеров, расположенных как можно дальше друг от друга. Работа алгоритма делится на несколько этапов:

1. Случайно выбрать  $k$  точек, являющихся начальными «центрами масс» кластеров.
2. Отнести каждый объект к кластеру с ближайшим «центром масс».
3. Пересчитать «центры масс» кластеров согласно их текущему составу.
4. Если критерий остановки алгоритма не удовлетворен, вернуться к п. 2.

В качестве критерия остановки работы алгоритма обычно выбирают минимальное изменение среднеквадратической ошибки. Так же возможно останавливать работу алгоритма, если на шаге 2 не было объектов, переместившихся из кластера в кластер.

К недостаткам данного алгоритма можно отнести необходимость задавать количество кластеров для разбиения.

#### Нечеткие алгоритмы

Наиболее популярным алгоритмом нечеткой кластеризации является алгоритм  $c$ -средних ( $c$ -means). Он представляет собой модификацию метода  $k$ -средних. Шаги работы алгоритма:

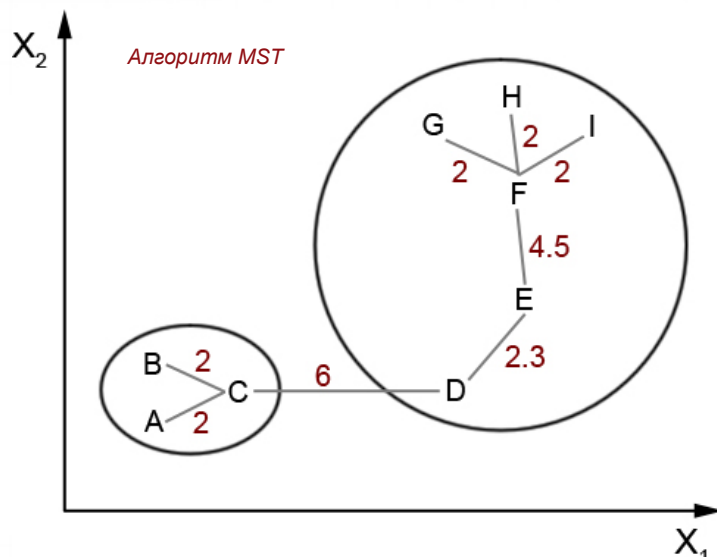
1. Выбрать начальное нечеткое разбиение  $n$  объектов на  $k$  кластеров путем выбора матрицы принадлежности  $U$  размера  $n \times k$ .
2. Используя матрицу  $U$ , найти значение критерия нечеткой ошибки:

$$E^2(X, U) = \sum_{i=1}^N \sum_{k=1}^K U_{ik} \|x_i^{(k)} - c_k\|^2$$

где  $c_k$  – «центр масс» нечеткого кластера  $k$ :

$$c_k = \sum_{i=1}^N U_{ik} x_i$$

4. Перегруппировать объекты с целью уменьшения этого значения критерия нечеткой ошибки.



5. Возвращаться в п. 2 до тех пор, пока изменения матрицы  $U$  не станут незначительными.

Этот алгоритм может не подойти, если заранее неизвестно число кластеров, либо необходимо однозначно отнести каждый объект к одному кластеру.

## Алгоритмы, основанные на теории графов

Суть таких алгоритмов заключается в том, что выборка объектов представляется в виде графа  $G=(V, E)$ , вершинам которого соответствуют объекты, а ребра имеют вес, равный «расстоянию» между объектами. Достоинством графовых алгоритмов кластеризации являются наглядность, относительная простота реализации и возможность внесения различных усовершенствований, основанных на геометрических соображениях. Основными алгоритмами являются: алгоритм выделения связанных компонент, алгоритм построения минимального покрывающего (остовного) дерева и алгоритм послойной кластеризации.

#### Алгоритм выделения связанных компонент

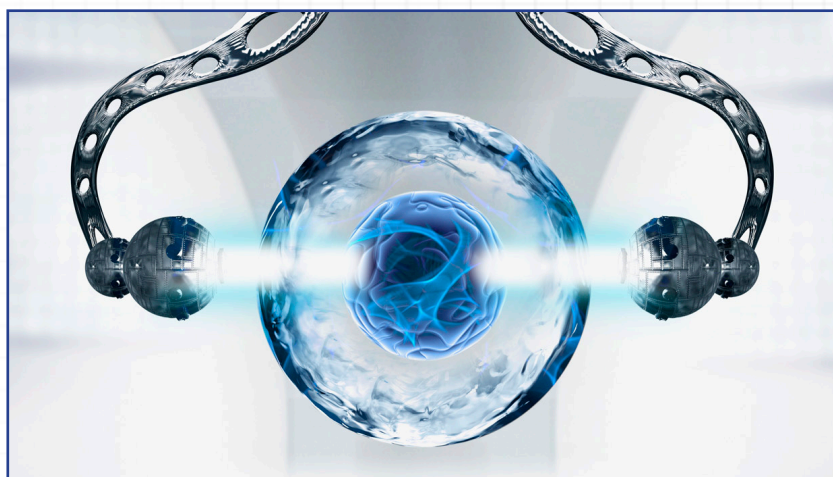
В алгоритме выделения связанных ком-

понент задается входной параметр  $R$  и в графе удаляются все ребра, для которых «расстояния» меньше  $R$ . Соединенными остаются только наиболее близкие пары объектов. Смысл алгоритма заключается в том, чтобы подобрать такое значение  $R$ , лежащее в диапазоне всех «расстояний», при котором граф «развалится» на несколько связанных компонент. Полученные компоненты и есть кластеры.

Для подбора параметра  $R$  обычно строится гистограмма распределений попарных расстояний. В задачах с хорошо выраженной кластерной структурой данных на гистограмме будет два пика – один соответствует внутрикластерным расстояниям, второй – межкластерным расстояниям. Параметр  $R$  подбирается из зоны минимума между этими пиками. При этом управлять количеством кластеров при помощи порога расстояния довольно затруднительно.

#### Алгоритм минимального покрывающего дерева

Алгоритм минимального покрывающего дерева сначала строит на графе минимальное покрывающее дерево, а затем последовательно удаляет ребра с наибольшим весом. На рисунке изображено минимальное покрывающее дерево, полученное для девяти объектов. ►



## Сравнение алгоритмов

Таблица 1. Вычислительная сложность алгоритмов

| Алгоритм кластеризации         | Вычислительная сложность                                   |
|--------------------------------|------------------------------------------------------------|
| Иерархический                  | $O(n^2)$                                                   |
| k-средних                      | $O(nkl)$ , где $k$ – число кластеров, $l$ – число итераций |
| c-средних                      |                                                            |
| Выделение связных компонент    | Зависит от алгоритма выделения связных компонент           |
| Минимальное покрывающее дерево | $O(n^2 \log n)$                                            |
| Послойная кластеризация        | $O(\max(n, m))$ , где $m < n(n-1)/2$                       |

Путем удаления связи, помеченной  $CD$ , с длиной равной 6 единицам (ребро с максимальным расстоянием), получаем два кластера:  $\{A, B, C\}$  и  $\{D, E, F, G, H, I\}$ . Второй кластер в дальнейшем может быть разделен еще на два кластера путем удаления ребра  $EF$ , которое имеет длину, равную 4,5 единицам.

### Послойная кластеризация

Алгоритм послойной кластеризации основан на выделении связных компонент графа на некотором уровне расстояний между объектами (вершинами). Уровень расстояния задается порогом расстояния  $c$ . Например, если расстояние между объектами:

$$0 \leq \rho(x, x') \leq 1$$

то:

$$0 \leq c \leq 1$$

Алгоритм послойной кластеризации формирует последовательность подграфов графа  $G$ , которые отражают иерархические связи между кластерами:

$$G^0 \subseteq G^1 \subseteq \dots \subseteq G^m$$

где  $G^t = (V, E^t)$  – граф на уровне  $c^t$ ,

$$E^t = \{e_{ij} \in E : \rho_{ij} \leq c^t\}$$

$c^t$  –  $t$ -ый порог расстояния,

$m$  – количество уровней иерархии,

$G^0 = (V, \emptyset)$ ,  $\emptyset$  – пустое множество ребер графа, получаемое при  $t^0 = 1$ ,

$G^m = G$ , то есть граф объектов без ограничений на расстояние (длину ребер графа), поскольку  $t^m = 1$ .

Посредством изменения порогов расстояния  $\{c^0, \dots, c^m\}$ , где  $0 = c^0 < c^1 < \dots < c^m = 1$ , возможно контролировать глубину иерархии получаемых кластеров. Таким образом, алгоритм послойной кластеризации способен создавать как плоское разбиение данных, так и иерархическое.

## Выбор наилучшего алгоритма

Выбор алгоритма кластеризации лежит полностью на пользователе. Остановиться на наилучшем сочетании «функция расстояния – алгоритм – функция межкластерного расстояния» возможно лишь проведя серию экспериментов. Первоначально можно лишь ограничить список мер и алгоритмов, отсеяв заранее непригодные в контексте решаемой задачи. ■

Таблица 2. Сравнительная таблица алгоритмов

| Алгоритм кластеризации         | Форма кластеров | Входные данные                                             | Результаты                                                  |
|--------------------------------|-----------------|------------------------------------------------------------|-------------------------------------------------------------|
| Иерархический                  | Произвольная    | Число кластеров или порог расстояния для усечения иерархии | Бинарное дерево кластеров                                   |
| k-средних                      | Гиперсфера      | Число кластеров                                            | Центры кластеров                                            |
| c-средних                      | Гиперсфера      | Число кластеров, степень нечеткости                        | Центры кластеров, матрица принадлежности                    |
| Выделение связных компонент    | Произвольная    | Порог расстояния $R$                                       | Древовидная структура кластеров                             |
| Минимальное покрывающее дерево | Произвольная    | Число кластеров или порог расстояния для удаления ребер    | Древовидная структура кластеров                             |
| Послойная кластеризация        | Произвольная    | Последовательность порогов расстояния                      | Древовидная структура кластеров с разными уровнями иерархии |

### Список литературы

1. Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекций. МГУ, 2007.
2. Jain A., Murty M., Flynn P. Data Clustering: A Review. // ACM Computing Surveys. 1999. Vol. 31, no. 3.
3. Котов А., Красильников Н. Кластеризация данных. 2006.
3. Мандель И. Д. Кластерный анализ. — М.: Финансы и Статистика, 1988.
4. Прикладная статистика: классификация и снижение размерности. / С. А. Айвазян, В. М. Бухштабер, И. С. Енюков, Л. Д. Мешалкин — М.: Финансы и статистика, 1989.
5. Информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных — [www.machinelearning.ru/](http://www.machinelearning.ru/)
6. Чубукова И.А. Курс лекций «Data Mining», Интернет-университет информационных технологий — [www.intuit.ru/departament/database/datamining/](http://www.intuit.ru/departament/database/datamining/)



# Сказ о том, как Bash и SVG спасли от рутины

**Автор**

Алексей Шаферов  
Нижегородская Группа  
Пользователей Linux  
(NNLUG)

Есть такой замечательный формат векторной графики – Scalable Vector Graphics, SVG. Чем же он такой замечательный? Ну, например:

1. SVG это открытый формат, он не является чьей-либо собственностью
2. SVG является подмножеством языка XML и, соответственно, он является текстовым
3. прекрасно интегрируется с HTML и XHTML
4. SVG совместим с CSS, что позволяет управлять отображением элементов с помощью таблиц стилей.
5. в SVG текст остается текстом, благодаря этому документы SVG могут индексироваться поисковыми машинами, также пользователи могут выделять и копировать текст.

Мне особенно нравится третий пункт списка. Что особенно интересно, возможно не только встраивать SVG документы в HTML, но и наоборот, встраивать в SVG целые документы HTML или отдельные теги. Это позволяет создавать очень интересные вещи, недоступные для Flash и SilverLight.

Ниже небольшой пример встраивания HTML в SVG, сайт находящийся внутри картины остается скроллируемым и кликабельным. К сожалению, не во всех браузерах это рабо-

тает одинаково хорошо, в Firefox'e проблем не обнаружено, только при прокрутке несколько притормаживает.

Я очень надеюсь, что со временем появятся сайты активно использующие SVG, потому что мне кажется, у такого подхода будут преимущества перед технологиями Flash и SilverLight. Например, появляется возможность использования CMS предназначенных для HTML, и кроме того, такие сайты будут без проблем индексироваться поисковыми системами.

Но сейчас не об этом, сейчас о пункте номер 2. Когда я работал дизайнером в одной маленькой фирме, нам поступил заказ на две партии дисконтных карточек, по тысячи штук в каждой партии. Тираж для фирмы небольшой, но задача сильно осложнялась необходимостью нумерации карточек, и на каждой карточке должен был присутствовать штрих-код, соответствующий номеру.

Фирма сидела на Corel DRAW x3. Средства автоматизации в нем присутствуют, и сделать карточки с номерами не проблема. Но вот как сделать штрих-коды по этому номеру никто из нас не знал (возможно, что есть решение средствами самого Corel'a, но найти не удалось).

Первым решение предложил директор:

- Ну, ничего, сядем и настулим по-быстрому.

Но мне такое решение показалось не достаточно оптималь- ➤





ным, тут-то я и вспомнил о формате SVG. Поскольку он является текстовым, то его можно без проблем редактировать программными средствами, например, из скрипта на bash. Я давно использую Linux в качестве основной домашней системы (а на данный момент Ubuntu единственная ОС на моем компьютере) и знаком с bash скриптами. К сожалению, на работе не дали ставить Linux, зато была возможность «сходить домой» по ssh. Основная графика карточек была экспортирована из Corel DRAW в формат PNG и вставлена в SVG целой картинкой, в местах предназначенных для номера и штрих-кода было оставлено пустое место. Конечно, можно было оставить в векторном формате, но по непонятным причинам при открытии .cdr в Inkscape'e (а также и при экспорте в SVG в Corel DRAW) некоторые элементы расползались. Для редактирования SVG использовался кроссплатформенный свободный векторный редактор Inkscape.

В Inkscape был подготовлен шаблон. Шаблон содержал лицевую и обратную стороны для четырех карточек, и линии разметки для удобства.

Немного о техпроцессе, для справки. Карточки печатаются на пластике (4 штуки на лист, обе стороны), лист пластика режется пополам, обе половинки складываются изображением от себя и дополняются листами ламината, после чего получившийся бутерброд отправляется в печку, а после готовности в специальную машинку для выбивания карточек (по принципу дырокола).

В местах, отведенных для номера были написаны номера 000000000001 - 000000000004 (на листе 4 карточки, поэтому 4 номера), а на месте для штрих-кода была вставлена пустая картинка. Почему пустая? Да, Inkscape умеет генерировать штрих-коды, но опять же не известно как этот процесс автоматизировать, поэтому для генерации штрихов пришлось использовать отдельную программку – Barcode. На выходе Barcode выдает Postscript файл, который с помощью утилиты convert (из пакета ImageMagick) конвертируется в PNG и вставляется в конечный SVG документ.

Для того чтобы из всего этого наконец-то получить 1000 визиток был написан небольшой скрипт, который я привожу в первоначальном виде (да, я знаю, что можно оптимизировать).

```
for n in `seq 100000000001 4 100000001000`
do
echo $n
nn=$n
fn=new-$n.svg
cp 6.svg $fn

barcode -b $nn -e EAN -u mm -g 31x17+0+0 -m 0
-E | convert -density 600x600 -trim - bcs/$nn-
bc.png

sed -i «s/01bc.png/bcs\/$nn-bc.png/g» $fn
sed -i «s/1000000000001/$nn/g» $fn

let «nn=$nn+1»

barcode -b $nn -e EAN -u mm -g 31x17+0+0 -m 0
-E | convert -density 600x600 -trim - bcs/$nn-
bc.png

sed -i «s/02bc.png/bcs\/$nn-bc.png/g» $fn
sed -i «s/1000000000002/$nn/g» $fn

let «nn=$nn+1»

barcode -b $nn -e EAN -u mm -g 31x17+0+0 -m 0
-E | convert -density 600x600 -trim - bcs/$nn-
bc.png

sed -i «s/03bc.png/bcs\/$nn-bc.png/g» $fn
```

```
sed -i «s/1000000000003/$nn/g» $fn
```

```
let «nn=$nn+1»
```

```
barcode -b $nn -e EAN -u mm -g 31x17+0+0 -m 0
-E | convert -density 600x600 -trim - bcs/$nn-
bc.png
```

```
sed -i «s/04bc.png/bcs\/$nn-bc.png/g» $fn
```

```
sed -i «s/1000000000004/$nn/g» $fn
```

```
done
```

Скрипт достаточно простой, но поясню что тут происходит. В цикле от 100000000001 до 100000001000 (наличие 10000000 объясняется заданием, т. е. именно такие нужны были номера) и с шагом 4 копируется исходный файл шаблона (строка 6). Как уже говорилось выше, каждый файл содержит 4 карточки, поэтому для каждого файла генерируется 4 штрих-кода, 4 раза заменяется номер и 4 раза заменяется название файла со штрих кодом. Штрих-код нужного размера в формате Postscript отдается программе convert, которая отрезает лишние белые поля и сохраняет результат в PNG формате, в отдельной папке (строка 7). Номера и имена файлов заменяются с помощью потокового редактора sed.

В результате работы скрипта в папке получилось 250 файлов готовых к печати. Это была первая партия. Вторая партия отличалась только картинкой, положение номера и штрих-кода не менялось, поэтому для получения еще 1000 дисконтов достаточно было заменить файл с картинкой, повторный запуск скрипта при этом не требовался.

Вот пример SVG файла, получаемого «на выходе».



# Каталог Веб-студий



## ADVANCED SYSTEMS

Страна: Россия  
Город: Москва  
Год основания: 2005  
Сотрудников: 6  
Сайт: [www.artwebstudio.ru](http://www.artwebstudio.ru)



## Интернет-Агентство Прометей

Страна: Россия  
Город: Ульяновск  
Год основания: 2006  
Сотрудников: 15  
Сайт: [www.pro-mt.ru](http://www.pro-mt.ru)



## Aniart

Страна: Украина  
Город: Киев  
Год основания: 2006  
Сотрудников: 8  
Сайт: [www.aniart.com.ua](http://www.aniart.com.ua)



## Интернет-студия WEBROVER

Страна: Россия  
Город: Самара  
Год основания: 1997  
Сотрудников: 19  
Сайт: [www.webrover.ru](http://www.webrover.ru)



## Forumedia

Страна: Россия  
Город: Москва, Тула  
Год основания: 1998  
Сотрудников: 20  
Сайт: [www.forumedia.ru](http://www.forumedia.ru)



## Комп-АС

Страна: Россия  
Город: Саратов  
Год основания: 2010  
Сотрудников: 20  
Сайт: <http://komp-ac.ru>



## iTrack

Страна: Россия  
Город: Москва  
Год основания: 2004  
Сотрудников: 35  
Сайт: [www.itrack.ru](http://www.itrack.ru)



## ООО «Дом сайтов»

Страна: Россия  
Город: Тула  
Год основания: 2007  
Сотрудников: 10  
Сайт: [www.reee.ru](http://www.reee.ru)



## Виртуальные технологии

Страна: Россия  
Город: Иркутск  
Год основания: 1999  
Сотрудников: 11  
Сайт: [www.virtech.ru](http://www.virtech.ru)



## Рекламное агентство «Артэйн»

Страна: Россия  
Город: Воронеж  
Год основания: 2008  
Сотрудников: 10  
Сайт: <http://artejn.ru>



## Дизайн-студия ElipseArt

Страна: Россия  
Город: Пермь  
Год основания: 2007  
Сотрудников: 6  
Сайт: [www.elipseart.ru](http://www.elipseart.ru)



## Центр репутационных технологий

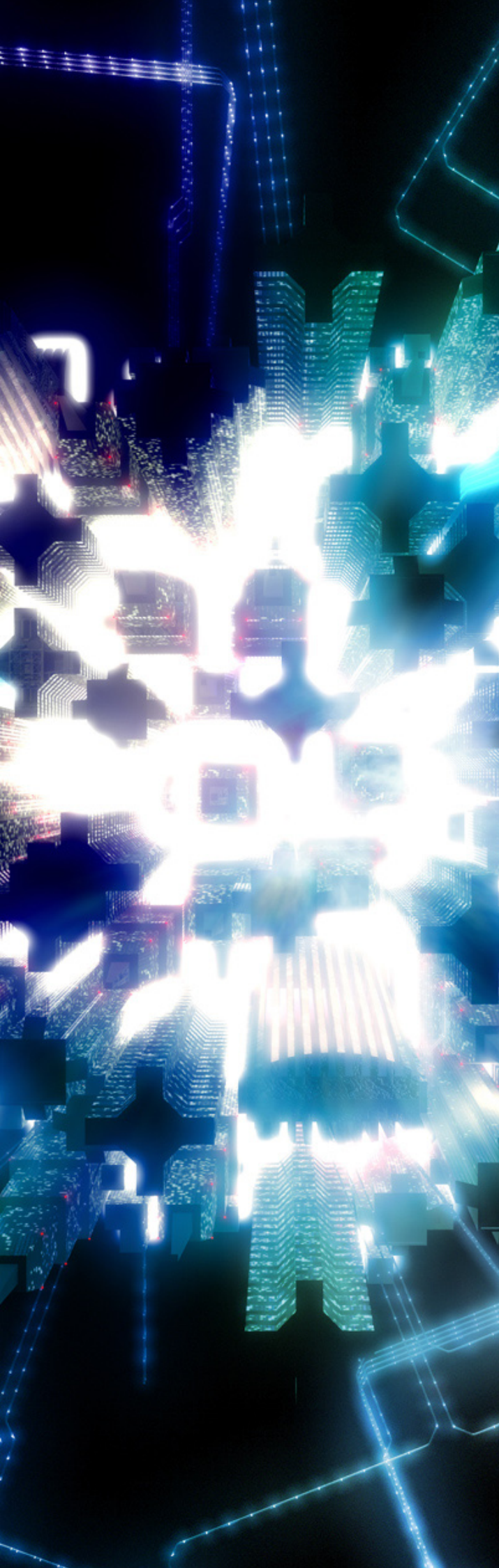
Страна: Россия  
Город: Владивосток  
Год основания: 2005  
Сотрудников: 8  
Сайт: <http://vcrt.ru>

**Добавить веб-студию в каталог журнала бесплатно**



**ДОБАВИТЬ**

Публикуемые в журнале компании взяты из каталога сайта издания. Данные о компаниях публикуются по мере их добавления в каталог. Добавить в каталог данные о компании может любая организация. Добавление данных в каталог сайта осуществляется бесплатно на добровольной основе самой организацией. Редакция журнала Веб-Аналитик.ИНФО не занимается добавлением данных в каталог сайта и не несет ответственности за предоставляемую информацию от организаций.



## ИНТЕРНЕТ

**Интервью с  
Дмитрием Агафоновым  
(Координатор и руководитель  
портала Ubuntu.ru )**

**Обзор Ubuntu 10.10  
Maverick Meerkat Alpha 3**

**Обновленный файловый  
менеджер Nautilus в Ubuntu 10.10**

**Исследование беспроводного  
интернета на вокзалах Москвы**

**Два дня из жизни IT-шника**





# Интервью с Дмитрием Агафоновым

## [ Координатор и руководитель портала Ubuntu.ru ]

Беседовал Станислав Горнаков

Операционная система Ubuntu с каждым новым выпуском доказывает свою состоятельность на рынке операционных систем, и все больше и больше привлекает поклонников в свой стан. Очевидно, что застоявшаяся гегемония Windows нравится далеко не всем и в особенности людям, связанным с IT технологиями. Мы решили развить эту тему и пообщаться с Дмитрием Агафоновым, который является координатором команды поддержки огромного русскоязычного сообщества пользователей Ubuntu.ru.

**Станислав Горнаков:** Здравствуйте, Дмитрий! Рады приветствовать вас на страницах журнала Веб-Аналитик.ИНФО. Дмитрий, давайте начнем с самой Ubuntu, ее истории и целях, а затем уже поговорим о русскоязычном сообществе пользователей Ubuntu. Может быть не все в курсе, что такое Ubuntu, поэтому расскажите кратко об истории возникновения данного дистрибутива, самой компании Canonical и ее целях.

**Дмитрий Агафонов:** Здравствуйте! Я также рад возможности рассказать немного об Ubuntu и о людях, которые работают над этим интересным проектом.

Обращаясь к истории компании Canonical и дистрибутива Ubuntu невозможно пройти мимо основателя и вдохновителя всего этого: Марка Шаттлворта (Mark Shuttleworth). Марк родился в Южно-Африканской Республике и получил образование в области Финансов и Информационных Технологий в Университете Кейптауна, где для него открылся мир программ с открытыми исходными кодами. Некоторое время Марк был одним из разработчиков Debian Linux и, работая в команде этого

заслуженно уважаемого и популярного дистрибутива, он понял как положительные, так и отрицательные стороны проекта Debian и решил что-то изменить. Так в 2004 году родились дистрибутив Ubuntu и компания Canonical. Главными целями в работе стали четкие сроки выхода (а не по графику «когда сделаем, тогда и выйдет») и готовность дистрибутива, как продукта, для массового потребительского рынка в качестве ОС общего назначения, а в последствии и как сервера. Компания Canonical, таким образом, стала официальным разработчиком Ubuntu, коммерческим OEM партнером для производителей компьютерной техники, а также стала оказывать техническую поддержку клиентам, установившим Ubuntu.

**С. Г.:** Размах компании впечатляет, на данный момент Canonical имеет офисы в США, Англии, Канаде и на Тайване, а также офис по месту регистрации компании на Острове Мэн. Кстати, почему именно Остров Мэн?

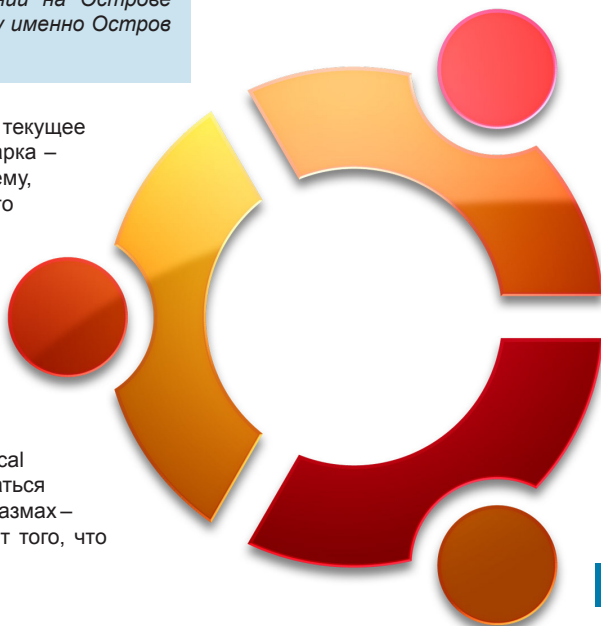
**Д. А.:** Основное текущее место жительства Марка – Остров Мэн. По-моему, вполне логично, что компания зарегистрирована там, не правда ли? А вообще, Остров Мэн – это интересное, как с географической и исторической, так и с политической точки зрения место.

При этом Canonical вряд ли может считаться крупной компанией. Размах – это скорее эффект от того, что

география деятельности охватывает весь мир: оказание техподдержки, общение с OEM производителями, разработчики. Настоящий размах – это то, что с проектами Canonical работает огромное сообщество людей вне компании, которые оценили гибкость предоставленных решений для работы над opensource проектами, а также Ubuntu, как мощную платформу для реализации своих идей.

**С. Г.:** Операционная система Ubuntu имеет несколько разных версий, охарактеризуйте, пожалуйста, каждую из них.

**Д. А.:** Изначально выпускалась только универсальная версия для настольных компьютеров – Desktop. Затем, с развитием дистрибутива, появилась серверная (Server) редакция, которая позволила многим системным администраторам гармонично дополнить рабо-



тающие на персональных компьютерах сотрудников системы, соответствующими серверными решениями с поддержкой самых новейших технологий. В последние годы особой популярностью стали пользоваться так называемые нетбуки, назначение которых и их технические параметры заставили разработчиков задуматься об оптимизации системы. В результате была выпущена специальная (Netbook) версия дистрибутива Ubuntu. Главные особенности версии для нетбуков – экономичность, удобство и скорость подключения к различным сетям, оптимизация для работы на малом экране.

**С. Г.:** Кроме этого, дистрибутив Ubuntu для настольных компьютеров имеет еще несколько разновидностей, таких известных как *Kubuntu*, *Xubuntu* и *Edubuntu* и менее известных *Gobuntu*, *Mythbuntu* и так далее. Расскажите об этих редакциях Ubuntu, в чем их различия и зачем они нужны.

**Д. А.:** Да, действительно, базовая версия дистрибутива Ubuntu разрабатывается и тщательно тестируется с пользовательским окружением GNOME. При этом для многих пользователей более соответствуют желаниям другие окружения. Главным образом это KDE, а также XFCE или LXDE.

Тут надо понимать, что Linux-дистрибутивы – это комплекты ПО, включающие ядро ОС, системное окружение, пользовательское окружение и прикладные программы. Таким образом, можно изменять эти комплекты, создавая различные версии дистрибутива для конкретных нужд. При этом на работающей системе могут быть одновременно установлены разные комплекты. Это потрясающая гибкость. Это то, что мы любим в Linux.

Итак, *Kubuntu* – это дистрибутив Ubuntu с пользовательским окружением KDE. *Xubuntu* и *Lubuntu* используют XFCE и LXDE, соответственно – это «легковесные» оболочки для компьютеров с небольшой производительностью.

Ориентировать Ubuntu для конкретных задач стало возможно благодаря изменению комплектов прикладного ПО. Так появились *Edubuntu* – дистрибутив для учебных заведений, *Mythbuntu* – развлекательный медиацентр, *Ubuntu Studio* – дистрибутив для творчества.

Еще есть множество версий и вариантов системы, которые создаются командами энтузиастов и коммерческими компаниями для собственных нужд.

**С. Г.:** Все версии Ubuntu бесплатны, соответственно возникает вопрос, на чем и как зарабатывает компания Canonical, ведь программисты тоже люди, они хотят кушать и их семьи тоже.



# CANONICAL

**Д. А.:** Основной заработок компании сейчас – это оказание технической поддержки, работа с OEM, а также ряд новых сервисов: платное расширение для удаленного хранилища Ubuntu One и магазин музыки, доступный там же. Руководство компании считает выбранный путь монетизации верным и продолжает в первую очередь развитие проекта Ubuntu, как главного козыря для привлечения клиентов на свои платные сервисы.

Однако, насколько мне известно, в настоящее время компания еще не вышла на прибыль, хотя и значительно приблизилась к этому. Таким образом, пока компания спонсируется самим Марком Шаттлвортом. Штат компании получает зарплату, а значит, у программистов и менеджеров есть на что покусаться, не волнуйтесь!

**С. Г.:** Как часто выходят обновления Ubuntu? Регулярны ли эти обновления или все зависит от настроения разработчиков?

**Д. А.:** Модель разработки открытых проектов приводит зачастую к тому, что программы быстро эволюционируют, меняются. Часто происходит смена лидера в том или ином классе пользовательских приложений. В открытых программах успешнее выявляются и исправляются ошибки. Это приводит к необходимости частых обновлений ПО, которое устанавливается в составе системы.

Работа над поддерживаемыми системами ведется постоянно. Обновления безопасности выходят для Ubuntu очень оперативно и так часто, как это

требуется. Однако в рамках одного релиза обновление версий программ обычно не производится в угоду совместимости и согласно золотому правилу администрирования: «работает – не трогай».

Раз в полгода выходят новые версии дистрибутивов семейства Ubuntu. Это необходимо для обновления на новые версии всего комплекта ПО в дистрибутиве, для внесения новшеств, оттачивания различных аспектов работы системы, интеграции новейших технологий.

Релизы нумеруются по правилу «2 цифры года, точка, месяц», текущая версия – это 10.04, следующая 10.10 выйдет в октябре.

**С. Г.:** Я сам конечно в курсе графика обновлений системы, так как являюсь поклонником этой ОС, но задал этот вопрос, потому что у большинства пользователей бытует мнение, что все Linux дистрибутивы – это дистрибутивы, собранные «на коленке». Что доверять им нельзя и когда-нибудь придет такое время, что разработчики забросят «свою игрушку», а мы останемся у «разбитого корыта».

**Д. А.:** Во-первых, Ubuntu поддерживается коммерческой компанией. Это бизнес и текущее состояние дел говорит о росте популярности системы и услуг, о хорошем будущем данного направления. Canonical, также как и другие коммерческие спонсоры Linux-дистрибутивов, гарантирует своим пользователям, что релизы будут поддерживаться определенное время (минимум 1.5 года для настольных систем ▶





и до 5 лет для LTS версий серверной редакции). Кроме того, уделяется особое внимание вопросу простой автоматической миграции на новые версии. Так что «у разбитого корыта» могут остаться лишь те, кто сам не захочет шагнуть в ногу с прогрессом в области open-source, предоставляемым компанией Canonical и тысячами разработчиков по всему миру.

Подключитесь к сети, устанавливайте бесплатно доступные для вас обновления безопасности, бесплатно обновляйтесь на новые версии дистрибутива и вы будете всегда иметь все самое лучшее из мира ПО с открытым кодом. А быть на острие прогресса – это здорово и воодушевляет!

**С. Г.:** Скажите, Ubuntu самодостаточная и сформировавшаяся система и она действительно может конкурировать с Windows?

**Д. А.:** Если коротко, то да. Как система, Ubuntu отлично справляется со своими задачами. Кроме того, уже сразу после установки дистрибутива обычно можно выполнять подавляющее число повседневных действий пользователя ПК. Работа в Интернет, почта, офисные документы, пасьянсы :)

Обычно проблемы возникают именно в тот момент, когда приходится сталкиваться с миром Windows. Специфическое прикладное ПО, протоколы взаимодействия с сервисами, форматы файлов. Вы не представляете, сколько сил приходится тратить разработчикам открытого ПО для того, чтобы открыть «окна»... Но и здесь уже очень впечатляющие результаты. Проблем все меньше и меньше.

**С. Г.:** Вы знаете, мне порой кажется, что Ubuntu для конкуренции с Windows не хватает каких-то мелочей и продуманности, например «нормальных» и привычных названий. Уже много так сказать «технического сленга». Для примера, вовремя установки пакета или обновлений, фраза плана «Идет настройка триггера...» думается, приводит простых пользователей в некоторое недоумение, что мол система-то для технарей. Может имеет смысл все же придерживаться привычных и понятных большинству пользователей названий?

**Д. А.:** Знаете, тут опять скорее проблема не столько в «техничности» сленга, сколько в его отличии от уже используемого в той самой, другой системе. Хотя, на самом деле, проблема есть и для решения предпринимаются определенные усилия. Марк в настоящее время сместился с должности управляющего компанией в сторону архитектора или, если хотите, дизайнера.

Была образована команда профессиональных дизайнеров, которые работают над развитием концепций более удобного и понятного пользовательского окружения для Ubuntu. Второй год подряд проходит акция «100 бумажных порезов», где совместно с пользователями производится поиск и устранение тех самых мелких недочетов, которые на самом деле очень портят впечатление от работы с системой. Результаты уже есть и многое еще в работе.

**С. Г.:** К слову, со стороны создается такое впечатление, что Ubuntu всеми силами пытается сделать некий свой клон Mac OS X. Если это так, то какой смысл в этом?

**Д. А.:** Сходство действительно есть. Но тут нет особого умысла именно скопировать интерфейс конкретной ОС. Просто в рамках задачи улучшения, упрощения, повышения интуитивности интерфейса приходится анализировать и примерять на себя уже созданные лучшие решения, пытаясь пойти дальше. Именно в данном состоянии находится работа над улучшением интерфейса в Ubuntu. На мой взгляд, Linux никогда еще не был так красив и удобен, как сейчас. А ведь работы еще ведутся!

**С. Г.:** Десятого октября выйдет новый релиз Ubuntu 10.10, что нам всем ждать от этого релиза?

**Д. А.:** Как минимум, каждый новый выпуск это, как я уже сказал, обновление версий всего ПО, которое входит в состав системы. А это приносит как исправление ошибок, так и порцию новых возможностей и функций. Например, обновление ядра расширит спектр поддерживаемого оборудования. Кроме того будет обновлено оформление и комплект ПО, который будет устанавливаться по умолчанию.

Готовясь к предыдущему выпуску, Ubuntu 10.04 LTS, который является релизом с долгосрочной поддержкой и должен быть, таким образом, наиболее стабильным, команде разработчиков пришлось «придержать» ряд изменений. Эти новшества и будут постепенно

внедряться в 10.10 и последующие выпуски. Уже были анонсированы изменения в интерфейсе варианта для нетбуков: оболочка «Unity», а также было объявлено о запуске проекта «Light» – облегченного варианта Ubuntu для сверхбыстрого запуска и работы вне и до основной операционной системы. Это, конечно, не все изменения. Следите за новостями, и узнаете все с подробностями.

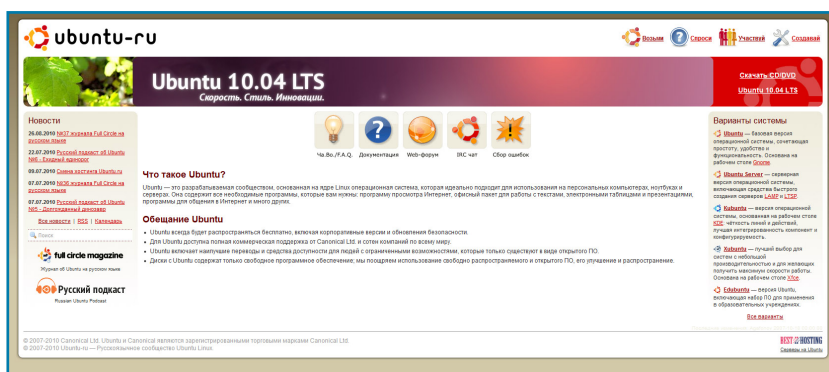
**С. Г.:** Дмитрий, давайте теперь поговорим о нашем сообществе и вас. Когда вы первый раз познакомились с Ubuntu, и как пришла мысль создать сообщество русскоязычных пользователей Ubuntu.ru?

**Д. А.:** Создать сообщество невозможно, оно образуется само там, где есть повод и интерес. Так, появившись в 2004 году, дистрибутив Ubuntu стал таким центром притяжения и формирования сообщества. В начале 2005 года был зарегистрирован домен и открыт русскоязычный форум. Именно как отображение факта появления сообщества, которое растет и развивается до сих пор.

Я же начал использовать Ubuntu в 2006 году, а в 2007 году вошел в команду поддержки ресурсов Ubuntu.ru. Как-то так получилось, что через некоторое время я стал координатором нашей команды. Наверное, у меня было больше всех свободного времени :)

**С. Г.:** Вы уже второй год подряд получаете аттестат действующей команды LoCo. Расскажите, что такое LoCo, что это за аттестат и какие преимущества он вам дает?

**Д. А.:** С самого начала работы над проектом Ubuntu в компании Canonical осознавали, какой важной частью открытых проектов является сообщество. Это не только пользователи и потенциальные подписчики платных услуг. Сообщество – это обратная связь, это генератор идей, это источник развития и «лакомая бумажка», определяющая правильность выбранных направлений движения. Фактически, все разработчики компонентов системы, переводчики,





люди, сообщающие об ошибках – все это тоже часть сообщества.

Для поддержки сообщества в Ubuntu был введен институт Локальных Команд (LoCo). Это группы энтузиастов, которые занимаются продвижением Ubuntu в своих городах и странах, организуют мероприятия, распространяют диски с системой и помогают новичкам освоиться. На больших территориях, таких как США и Россия действуют глобальные команды, поддерживающие локальные группы активистов. Такой командой и является наша *Russian Ubuntu LoCo Team*.

Каждые 2 года происходит процесс переаттестации команд, рассмотрение их деятельности и результатов работы. Подтверждающий статус (Approved LoCo Team) – это признание заслуг всех людей из нашей команды, которые помогают развивать проект, это доверие использовать торговые марки компании Canonical в работе наших ресурсов, это официальный статус ресурсов поддержки сообщества, это определенная ответственность как перед компанией, так и перед пользователями, которые ожидают определенного уровня поддержки на наших ресурсах.

Кроме того, на локальном уровне именно активно действующим командам оказывается больше внимания при поддержке для организации мероприятий, рассылке дисков и раздаточных материалов.

**С. Г.:** Сколько активных пользователей на данный момент насчитывает портал Ubuntu.ru.

**Д. А.:** Точные цифры назвать довольно сложно, у нас несколько десятков тысяч зарегистрированных пользователей, постоянно посещающих наши ресурсы. Кроме того, сотни тысяч посетителей ежемесячно просматривают 2-3 миллиона страниц. Информацию мы не скрываем, регистрацию для доступа не требуем. Так что можно сказать, что наша пользовательская база как минимум составляет несколько сотен тысяч русскоговорящих владельцев компьютеров по всему миру, которые интересуются Ubuntu, уже установили систему или планируют это сделать.

**С. Г.:** Что полезного есть на портале для простого пользователя?

**Д. А.:** Информация, новости, но главное поддержка. Если у кого есть вопросы по каким либо аспектам работы системы, то можно найти ответ через поиск или задать вопрос в подходящем разделе *форума*. Быстро решить проблему помогут в IRC или Jabber чате. Более фундаментальные описания и документация содержится в нашей *справочной wiki*.

Кроме того, люди со схожими инте-



ресами всегда найдут темы для обсуждений, и не обязательно околокомпьютерных.

**С. Г.:** А что в целом делает ваша команда для Ubuntu ОС и что может сделать простой пользователь, чтобы быть полезным для сообщества?

**Д. А.:** Группа перевода на русский язык делает, пожалуй, самую заметную работу. Это то, с чем непременно столкнутся все, кто использует Русский, как язык системы. Редактуры оформили большое количество статей в нашей русскоязычной документации. Периодически выходят переводы журнала *Full Circle*, а также *подкаст*. Тем не менее, техническая поддержка ресурсов, модерирование форума и чатов, тестирование и работа над ошибками системы – это то, что мы также делаем ежедневно и зачастую незаметно для большинства пользователей.

Простые пользователи – это те, кто

когда-то еще не были пользователями, а завтра могут стать «гуру». Главное, что можно делать – помогать другим: если вы знаете ответ на вопрос поднятый на форуме – напишите. Сообщайте модераторам, если где-то нарушаются правила и нормы поведения. Если вы нашли ошибку в переводе или в работе системы – сообщайте о них. Если вы умеете что-то и хотите помочь нашей команде, посмотрите *специальную страницу*, где мы ищем помощь. Если у вас есть идеи – *выскажите их*.

Все это поможет сделать систему и наши ресурсы лучше и полезнее. Сделает дистрибутив Ubuntu и сообщество такими, какими их хотим видеть все мы.

**С. Г.:** Дмитрий, спасибо за интервью, желаю вам и Ubuntu ОС всех благ!

**Д. А.:** Вам спасибо за интересные вопросы, Станислав. Удачи! ■



# Обзор Ubuntu 10.10 Maverick Meerkat Alpha 3

Автор

Дмитрий Канн  
Амерсфорт  
Нидерланды  
<http://dk-netherlands.blogspot.com>

5 августа, вышла третья альфа-версия Убунту 10.10 – Maverick Meerkat Alpha 3, вобравшая в себя функциональность, подготавливаемую для следующей стабильной версии системы, которая выйдет 10 октября этого года (10.10.10).

## Обновление с Ubuntu 10.04 LTS

Обновиться до третьей альфы с Ubuntu 10.04 LTS на десктопной версии можно с помощью команды:

```
update-manager -d
```

Запустившийся Update Manager сообщит о наличии новой версии 10.10. Нажмите **Обновить** и следуйте указаниям программы.

Чтобы обновить серверную Ubuntu 10.04 LTS, необходимо:

- Установить пакет update-manager-core (если он еще не установлен);
- Отредактировать /etc/update-manager/release-upgrades и установить Prompt=normal;
- Запустить менеджер обновлений командой `sudo do-release-upgrade -d`;
- Следовать указаниям программы.

## Что нового в Maverick Meerkat



### Обновление пакетов

Как обычно бывает при подготовке новой версии, пакеты (приложения и прочее ПО) обновляются очень часто. Многие из этих пакетов поступают автоматически при синхронизации с веткой Debian Unstable. В данный момент период автоматической синхронизации окончен, поэтому теперь изменений будет не так много.



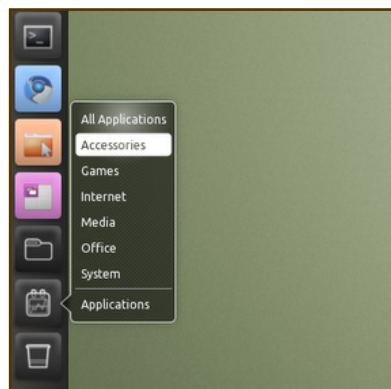
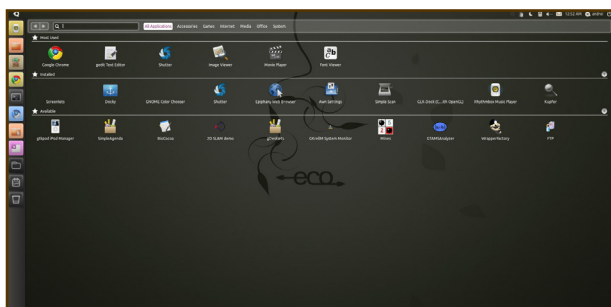
### Ubuntu

GNOME обновлен до текущей версии 2.31. Она, в частности, включает API dconf и gsettings. Evolution обновлена до версии 2.30, и работает теперь намного быстрее по сравнению с версиями, вошедшей в Ubuntu 10.04 LTS.



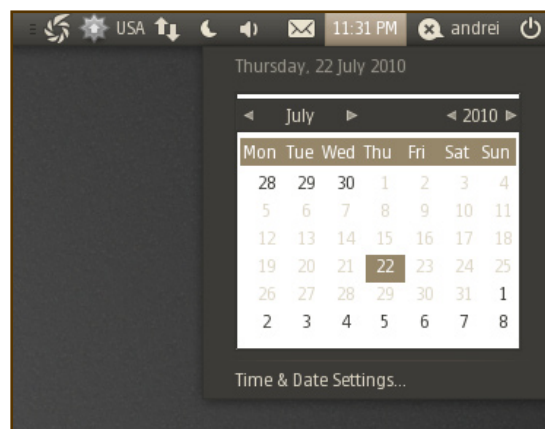
### Ubuntu Netbook Edition

По умолчанию оболочкой в нетбучной редакции Убунту теперь является Unity.

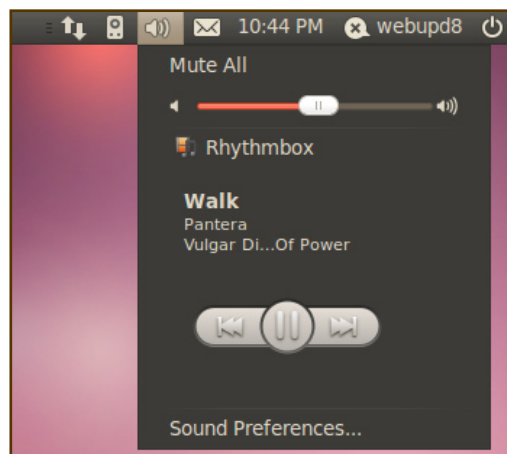


Среди прочего, она содержит глобальное меню приложения.

Индикатор даты/времени теперь предоставляет настоящий виджет календаря:



Также стоит отметить и редизайн меню регулятора громкости, впервые появившийся в прошлой альфе: ►





### Kubuntu

KDE обновлена до 4.5 release candidate. Веб-браузер по умолчанию теперь Rekonq, основанный на Webkit. Qt обновлена до текущей версии 4.7 beta.



### Xubuntu

Xfce4 обновлена до текущей версии 4.6.2, в которой исправлено множество багов и обновлены многие программы, использующиеся в Xubuntu. Новые программы по умолчанию: медиаплеер Parole вместо Totem Movie Player, программа для записи дисков Xfburn вместо Brasero, менеджер процессов xfce4-taskmanager вместо Gnome-Task-Manager.



### Edubuntu

Edubuntu наследует все изменения из Ubuntu. Пакет edubuntu-artwork был разделен на несколько меньших пакетов, чтобы упростить установку обновлений и уменьшить размеры загружаемых файлов.



### Software-Center

На стартовой странице Software-Center появились разделы **Лучшее (Featured)** и **Новое (What's New)**. Приложение стало работать быстрее, интерфейс стал отзывчивее, добавлена функция **История (History)**, где представлен список того, что было установлено, удалено и обновлено. Добавлена поддержка расширений (плагинов), по умолчанию специфические технические записи не отображаются.



### Ядро 2.6.35

Alpha 3 поставляется с ядром 2.6.35-14.19, основанным на финальной версии 2.6.35. Это ядро содержит ряд улучшений, связанных с безопасностью. Одно из самых важных нововведений изменяет стандартное поведение PTRACE, используемого gdb, strace, ltrace и т. д. Начиная с Ubuntu 10.10, только дочерние процессы могут быть отслежены, что обусловлено исходным значением 1 в /proc/sys/kernel/ptrace\_score. В некоторых системах такое значение может вызывать проблемы, например, в системах, используемых для разработки, и серверах, где имеются только администраторские учетные записи. Если использование sudo для PTRACE нежелательно, необходимо установить это значение в 0.



### Образы Ubuntu Server Cloud

Cloud-init – настраиваемый процесс инициализации облака Ubuntu Server, приобрел новые возможности еще в Maverick

Alpha 2, среди которых pluggable hooks, ebsmount, поддержка ext4 и новые параметры в формате cloud-config.

Начиная с Alpha 3, экземпляры облака могут управлять собственным ядром, включая обновление ядра с помощью apt. Это делается с помощью предоставляемой Амазоном утилиты pv-grub.



### Установка

Теперь при установке можно использовать новую файловую систему btrfs (при ручном разбиении), если раздел /boot поместить на другой системе.

## Известные проблемы

На данной стадии подготовки к релизу является закономерным наличие некоторых известных проблем, с которыми пользователи могут столкнуться в Maverick Alpha 3. Здесь собраны описания таких проблем, а также возможные исправления их.

- Установка в режиме OEM завершается неудачно при отсутствии подключения к Интернету, вследствие неудачной установки пакета oem-config из установочного образа. Если вам это требуется для тестирования, обеспечьте постоянное подключение к Сети в течение всей установки.

- Установщик для Windows Wubi завершается со сбоем. Проводится изучение проблемы.

- Ubuntu Netbook Edition с оболочкой Unity в настоящее время не предоставляет обходной возможности для систем с видеодрайвером не поддерживающим 3D-ускорения, как, например, NVidia.

- В системах с малым количеством памяти (256 МБ или меньше), ureadahead вызывает ошибки недостатка памяти, что может привести к невозможности загрузки.

- Приложения, использующие sqlite3, сталкиваются с множеством проблем, связанных с производительностью и правильностью выполнения запросов, например, banshee.

- Предрелизная версия Eucalyptus 2.0 включена в состав Alpha 3, хотя в ней до сих пор имеется ряд проблем. Например, регистрация после установки работает ненадежно, сложные топологии (такие, например, как отделяющие CLC от Walrus) пока не поддерживаются. При установке узловой (node) системы будьте готовы получить две проблемы с grub, которые могут быть проигнорированы без последствий.

- KDM выдает таймаут при загрузке с образа Kubuntu Live, после чего запуск отказоустойчивой X-сессии должен быть возможен. ■





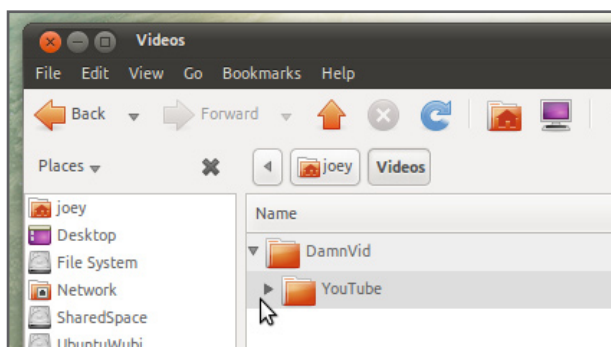
# Обновленный файловый менеджер Nautilus в Ubuntu 10.10

Автор

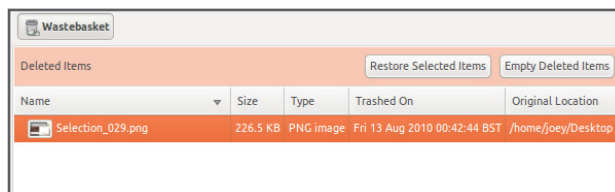
Дмитрий Канн  
Амерсфорт  
Нидерланды  
<http://dk-netherlands.blogspot.com>

В Ubuntu 10.10 Maverick Meerkat была принята из анпстри-ма новая версия Nautilus, файлового менеджера по умолчанию. Теперь, после прекращения добавления в Маверик функциональности («feature freeze»), можно ознакомиться с представленными изменениями.

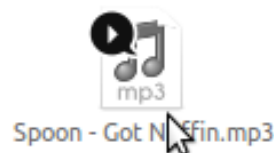
Помимо множества багфиксов, Наутилус в следующей версии Убунту характеризуется следующими, как это называется в научных кругах, плюшками.



- Разворачивание и сворачивание папок в списках клавишами «+» и «-».
- Пароли в сеансах теперь по умолчанию сохраняются.
- Надпись **File Browser** больше не добавляется к заголовку окна.
- Значки папок теперь используются как значки окна в режиме просмотра и пространственного просмотра (**spatial mode**).
- При просмотре содержимого **Корзины** в режиме списка, отображаются столбцы **Дата удаления (Trashed On)** и **Исходное расположение (Original Location)**.



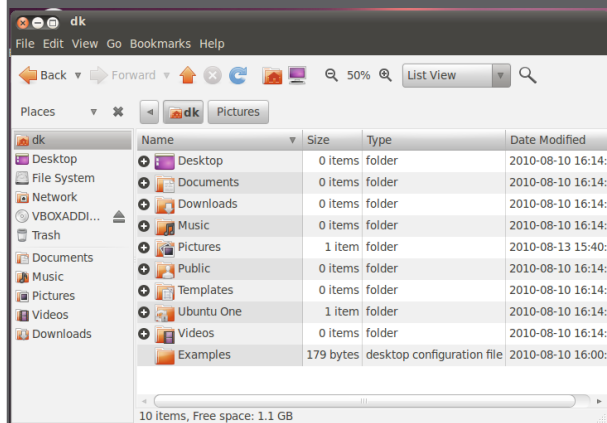
- «Вырезанные» файлы отображаются теперь с полупрозрачным значком.
- Размер эскиза по умолчанию может отныне быть задан.
- Новый диалог, отображаемый при возникновении конфликтов при выполнении операций копирования или перемещения файлов.
- В **Корзине** отображается кнопка восстановления выделенных файлов.
- Новый значок для предварительного прослушивания звуковых файлов:



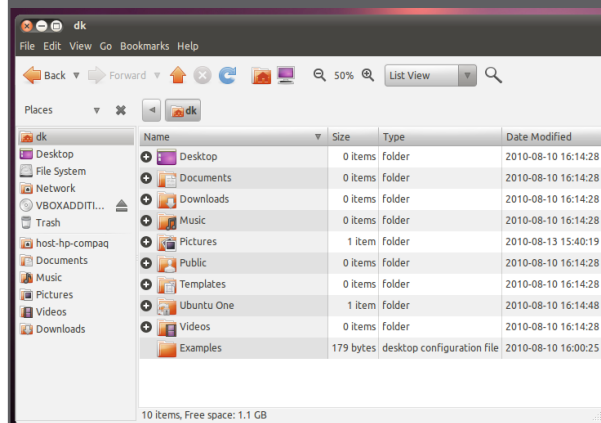
Воспроизведение останавливается, если укатить мышью значка.

Пункт **Отмонтировать (Unmount)** не отображается, если имеется пункт **Извлечь (Eject)** или **Безопасное извлечение (Safe Removal)**. ■

Скриншот со свежей версии Наутилуса. Видно, что по умолчанию, как минимум, шрифты те же, что и раньше



То же самое со шрифтом UbuntuBeta

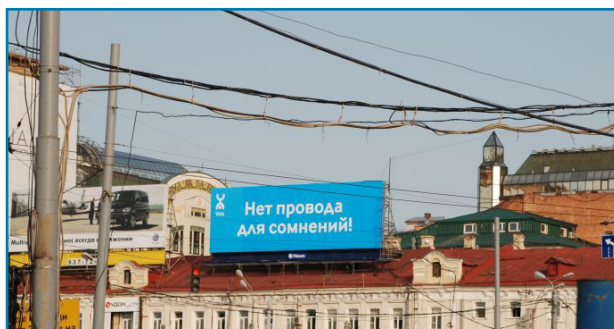


# Исследование беспроводного интернета на вокзалах Москвы

Автор

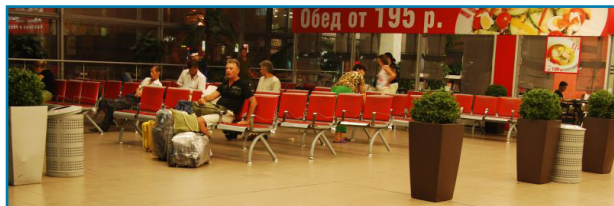
Олег Данилов  
Руководитель проекта  
gToday.ru

Зайнтересовавшись темой мобильного интернета, мы подготовили исследование девяти вокзалов Москвы. Все они расположены в пределах третьего транспортного кольца и практически равномерно распределены по его окружности, что даст гораздо более интересную картину.

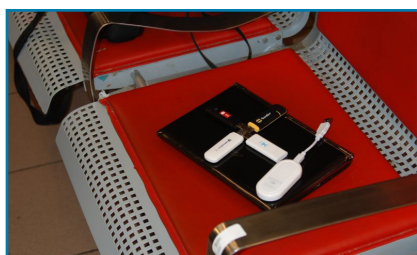


Не смотря на этот оптимистичный лозунг от Yota, окруженный проводами и кабелями (площадь Белорусского вокзала), мы решили все же усомниться и перед поездкой вновь открыли карты покрытия операторов, чтобы узнать, что нам обещают. Все кроме Билайна, который не имеет карты покрытия 3G по Москве, обещали нам высокоскоростной интернет на вокзалах.

Проводя исследование, мы приезжали на вокзал, сидели в одном из свободных залов ожидания поближе к окну, что, кстати, было кое-где не так легко – ожидающих слишком много, доставали нетбук, модемы операторов и начинали измерения под пристальным взглядом окружающих.



Это типичный зал ожидания. Курский вокзал. От многих других вокзалов его отличают мягкие вставки на креслах. Используемый при тестировании нетбук абсолютно «чистый», не имеющий никакого дополнительно установленного программного обеспечения, кроме ПО идущего в комплекте с модемами.



Модемы и приложения к ним покупались и устанавливались исключительно от операторов, чтобы избежать обвинений в неправильной настройке «левого» решения.

Павелецкий вокзал был первым в нашем долгом путеше-

ствии по Москве. Он один из самых загруженных вокзалов – отсюда ездит аэроэкспресс в Домодедово, электрички каширского направления, поезда дальнего следования в Поволжье и частично на Кавказ. Тут мы провели больше всего времени и сделали аж три замера в трех разных его местах.

Первой дислокацией был выбран зал ожидания на втором этаже. Результаты тестирования входящей скорости были следующими:

|            | Beeline | Comstar  | MegaFon  | MTS    | Yota     |
|------------|---------|----------|----------|--------|----------|
| Павелецкий | 710,32  | 2 140,47 | 6 504,07 | 707,03 | 1 226,52 |

Все скорости указаны в Кбита/сек

Затем мы переместились в «VIP зал», находится он ровно в том же опенспейсе что и обычный, огорожен ленточкой и отличается от обычного зала ожидания наличием диванов, телевизора, барышней собирающей деньги за вход (150 руб/час) и неработающими розетками. Кстати, за то, что мы хотели зарядить ноутбук, по тарифам «VIP зала» с нас хотели взять еще 60 рублей. «Насладившись» вдоволь статусом VIP, было принято решение не тратить больше нигде на это время и денег. Стоит ли говорить, что результаты тестирования были ровно такими же, как и в обычном зале.

Следующая дислокация была в зале ожидания аэроэкспресса. Он находится на первом этаже и там проходит довольно большой пассажиропоток. WiMAX тут уже не брал, Мегафон и Билайн держались очень хорошо, обеспечивая скорость выше 1 Мбита, у МТС дела были хуже. В этом зале ожидания так же есть платный Wi-Fi от Билайна и Мегафона. Розетку мы нашли в кафе расположенном прямо в зале ожидания. Столиков с розетками там несколько.

Далее последовала монотонная поездка по Москве. В результате мы пришли к следующим результатам

|               | Beeline  | Comstar  | MegaFon  | MTS      | Yota     |
|---------------|----------|----------|----------|----------|----------|
| Белорусский   | 90,65    | —        | 7 095,34 | 225,85   | 1 366,94 |
| Казанский     | 636,82   | 926,87   | 790,90   | 1 512,29 | 2 278,39 |
| Киевский      | 1 626,84 | —        | 156,83   | 33,42    | 375,15   |
| Курский       | 1 262,83 | 1 491,84 | 7 017,54 | 143,36   | 387,97   |
| Ленинградский | 636,18   | 1 169,16 | 1 401,05 | 435,02   | 1 766,98 |
| Павелецкий    | 710,32   | 2 140,47 | 6 504,07 | 707,03   | 1 226,52 |
| Рижский       | 1 035,60 | 774,07   | 75,28    | 1 499,53 | 1 657,17 |
| Савеловский   | 1 011,38 | 650,01   | 1 383,48 | 1 209,93 | 1 045,41 |
| Ярославский   | 1 012,31 | 1 125,18 | 1 658,03 | 1 520,19 | 1 620,25 |

Все скорости указаны в Кбита/сек

Из этой таблицы мы попытались отранжировать операторов по средней скорости. Что у нас получилось:

1. **Мегафон** – 2898 Кбит/сек.;
2. **Yota** – 1302 Кбит/сек.;
3. **Comstar** – 919 Кбит/сек.;
4. **Билайн** – 891 Кбит/сек.;
5. **МТС** – 809 Кбит/сек.

Следует помнить, что скорость очень сильно может зависеть от загруженности базовых станций оператора. И, возможно, в другой день операторы в каждом конкретном месте будут показывать совсем разные показатели. Так же у нас большие сомнения по поводу того, что Комстара нет в двух вокзалах. Хотя служба техподдержки нам так и не смогла помочь по телефону, упорно отправляя в сервис центр. ■



Марат Сибгатулин (eucariot)  
<http://eucariot.li.ru>

Все, что нам нужно было сделать, ►





это взять кабель оператора, обжать его, воткнуть в коммутатор 2, с которого он уйдет на вай-фай. Но не тут-то было. Сначала нас ждал утомительный подъем.

На улице 28 градусов, я в сандалиях (кто же знал, что выдвигаемся сегодня). Горнолыжная трасса порою под 45 градусов уносилась вверх. Наверху выясняется, что оператор сэкономил и отдал нам новый канал E1 в том же физическом кабеле, что и предыдущий. Таким образом, для обеспечения связи одному абоненту, придется временно отключить другого. Для минимизации времени простоя было решено сделать Y-образный разветвитель. То есть кабель от оператора будет воткнут в розетку RG-45, из которой выходят два провода.

С собой тогда мы взяли все, кроме кроссового ножа, и пары в розетку пришлось забивать обычным канцелярским ножом. Когда Y был готов, мы вытащили кабель оператора, быстро переобжали его в соответствие со стандартом Ethernet и вставили в Y, а два оставшихся хвоста в FOM и в конвертер. Порт FOM'a поднялся, а конвертера нет. Через три минуты с помощью светодиода мы поняли, что обжали в один конец 4 приема, а в другой 4 передачи. Быстро перезабили пары в розетке и оба порта поднялись. Отлично – физическое соединение у нас было.

Для теста подключаемся к порту Ethernet конвертера, настраиваем нужный IP, но связи нет. При этом второй канал точно поднялся – интернет есть. Что ж, исключаем слабые места, то есть переобжимаем и перезабиваем с этой стороны, просим также переобжать кабель со стороны маршрутизатора1. Не помогло.

Звонок оператору. Через минут 20-30 наконец выяснилось, что у них где-то был лишний loopback. После этого канал поднялся. Вот казалось бы и все: поставить сконфигурированный маршрутизатор и заменить коммутатор (старый DLink с завидной периодичностью подвисал и лишал народ даже локаль-



ной сети). Но беда не приходит одна. Оглушительный хлопок и тишина... Четыре человека замерли в молчании. FOM погас, а по помещению разнесся запах горелого железа. Вскрытие показало вздувшийся конденсатор. Очевидно, сгорел блок питания, и скорее всего это произошло из-за отсутствия

заземления.

Ситуация критическая – до конца рабочего дня нужна связь. Представитель абонента уносится (в прямом смысле, потому что спускаться шагом практически невозможно) вниз за резервным FOM'ом. Мы тем временем думаем над резервной схемой. В рас-

## СЛУШАЕМ ПОДКАСТЫ НА ВЕБ-АНАЛИТИК.ИНФО

[www.web-analitik.info/podcast/](http://www.web-analitik.info/podcast/)

- РУНЕТОЛОГИЯ с Максимом Спиридоновым
- ВРЕМЯ НОВОСТЕЙ с Владом Филатовым и Сергеем Болисовым
- PODCAST9 с Петром Диденко и Михаилом Черномордиковым
- APPLE МАНИЯ с Владом Филатовым и Сергеем Болисовым
- UBUNTU с Валентиной Мухамеджановой (Ubuntu.ru)
- Новости Hi-tech
- Новости IT



ДОБАВИТЬ СВОЙ  
ПОДКАСТ



поряжении только оптика до узла внизу горы, коммутатор DLink и два конвертера (старый, который использовался здесь прежде, и новый). Медиаконвертеров в наличии не было, но по оптике был подключен коммутатор.

Из-за сложности ситуации включили канатно-кресельную дорогу и представили с запасным FOM'ом вернуться быстро. Но, не судьба. Новый FOM никак не хотел понимать оптику и индикаторы NE и FE (статус локальной и удаленной стороны) горели красным. Попытки изменить конфигурацию через консоль тоже не возымели результата.

В итоге мы пришли к реализации плана Б: подключение конвертера к DLink'у и настройка на нем vlan. Казалось бы, чего проще и почему нельзя было реализовать эту схему изначально? Но беда не приходит одна, и не приходит она и вдвоем. До позднего вечера мы пытались это настроить, но никак не удавалось — с той будки наверху горы мы никак не могли увидеть местный маршрутизатор. В итоге проблема была локализована — не работал транк между двумя коммутаторами. Мы даже пробовали создавать полностью новый vlan, но безуспешно. С тех пор я ненавижу DLink'и.

Три разных устройства, три абсолютно разных интерфейса. Причем, как веб, так и телнет. На многих из них нет консольного порта, и если забудешь IP адрес, то подключиться без потери существующей конфигурации, скорее всего, не получится, они постоянно зависают и даже не могут договориться друг с другом. Предложение перевести блок питания со «здорового» FOM'a на «больной» клиентом было сразу отвергнуто, мол «еще и этот сгорит».

Канатно-кресельная дорога еще работает, так как должна забрать отсюда охранников, а нам в голову приходит еще одна идея. Пробуем ее реализовать, а тем временем охранники

уехали, оставив ключи от будки нам. Последняя попытка также ничего не дала, дорога уже остановилась — вниз тоже пешком. В ожидании разъяснений от старшего специалиста отдела ИТ, я вышел покататься в креслах, дверь хлопнулась с ключами внутри и нами снаружи. Ясности не стало и после звонка специалиста. Похоже, что два DLink'a не могли работать в паре.

Что ж, ничего не оставалось кроме как спуститься вниз, попытаться найти другой коммутатор и настроить его. Время около 10, голова уже не варит. Найденный DLink (опять же DLink, словно корпоративная этика заставляет их использовать) настроен на скорую руку. Завтра утром эпопея продолжится. Я записал на бумажке что и куда нужно будет воткнуть человеку, который поднимет его наверх.

## День второй

Утром следующего дня с первой попытки появилась связь до обоих клиентов, но проблема с большими пакетами. Интернета на старом канале не было: большие пакеты терялись в 80% случаев. Второй канал (новый) при этом работает нормально, нормально работал прежде и старый. Методом исключения приходим к тому, что проблема с физическим контактом, тем более что забивали розетку обычным ножом. Инструмента нормального здесь нет, но зато в 15 км от нас сейчас тот парень, что вчера забивал эту розетку. И он с инструментами — удачное стечение обстоятельств. На этот раз повезло и с дорогой, ее снова согласились включить для него. В течение получаса связь полноценно восстановилась, но то, ради чего все это затевалось, пока так и не работало. Была какая-то совершенно необъяснимая проблема с маршрутизацией и работой cisco. В тот момент пришлось отказаться от маршрутизатора на удаленной стороне. Та-

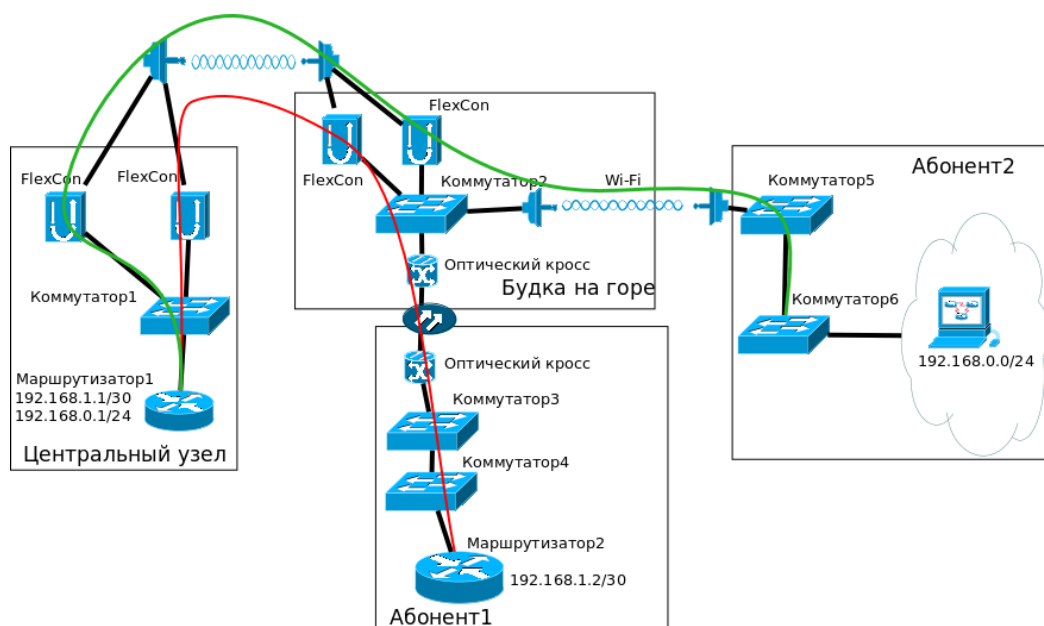
ким образом, схема подключения стала как на рисунке ниже.

Я оптимист и думал, что через неделю уже буду вспоминать об этом с улыбкой. Но не тут-то было. Беда приходит просто толпой. Прошла неделя стабильной работы, а потом аварийное отключение питания в серверной, и после перезагрузки каналы поднимаются с той же проблемой больших пакетов. Несколько часов мучений, курений, игр со скоростью и дуплексом, регистрация двух сопутствующих проблем, но никакого результата. Следующим утром прямым рейсом вылетаем на место аварии с инструментом и представителем оператора. Мучения продолжают.

Проверка кабеля, конвертера, пробовали включиться напрямую в кросс. Качество канала прыгает — то все нормально, то невероятные потери. И вдруг в организованную телефонную конференцию между нами и оператором «вырывается» новый человек, и сообщает, что на этом направлении у них авария. Так-то. Через пару часов исправляют, и в скором времени канал поднимается. Мы полны радости, что проблема не с нашей стороны и одновременно полны злости на такое отношение к нам оператора. Но радость была недолгой — после еще одной перезагрузки конвертера снова начались потери. Грешить на этот раз на физическое соединение не стали, а сразу обратились к тем, кто предоставляет канал. Снова долгая эпопея, и я расскажу лишь, чем все завершилось.

\*\*\*

Полноценно локализовать проблему так и не получилось, то ли она на оборудовании сотовой компании, то ли в наших конвертерах, но решением ее пока является удаление старого потока на коммутаторах оператора и добавление нового. Работы по определению истинной причины этой проблемы пока впереди. ■





# ХОСТИНГ



## Супер Старт —

полноценный хостинг начального уровня,  
идеально подходящий для размещения  
вашего первого сайта.

1 сайт, 1 база данных MySQL, 1 Гб дискового  
пространства, 10 почтовых ящиков  
с антивирусом и антиспамом, PHP, SSI, CGI,  
круглосуточная тех. поддержка.

# 99 Р/мес.\*

+ домен  
в подарок \*



\* При подписке на 1 год

(495) 508-99-59  
[www.rusonyx.ru/superstart](http://www.rusonyx.ru/superstart)





# Куда и кому стучит Google Chromium

Автор

Алексей Ужва  
Аспирант ВГТУ  
г. Волгоград  
www.uzhva.ru

**G**oogle обладает большим количеством великолепных сервисов, качество которых воистину впечатляюще, однако его мания слежки за всем и вся зачастую переходит границы. В этой небольшой статье речь пойдет о браузере Chromium.

Итак, многим из нас нравится браузер Chrome. Но далеко не всем нравится его политика конфиденциальности, которая позволяет использовать практически любые сведения о просматриваемых вами сайтах «в целях улучшения качества сервисов Google». Однако теперь у нас есть Chromium! Он Open Source, а значит шпионского кода там быть не должно. Эта догадка подтверждается английской википедией, говорящей что разница между Chrome и Chromium в отсутствии гугль-брендинга, автообновления и, ура, отсутствии отправки «шпионских» данных в Google!

## Давайте посмотрим, так ли это на самом деле

Итак, запускаем сниффер, запускаем браузер, и.... вы уже догадались, не правда ли? Перво-наперво браузер лезет на Google получить идентификационную куку (заметьте, я не произвожу никаких действий в интерфейсе):

```
HEAD / HTTP/1.1
Host: www.google.com
Connection: keep-alive
Content-Length: 0
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US)
AppleWebKit/534.6 (KHTML, like Gecko)
Chrome/6.0.489.0 Safari/534.6
Accept-Encoding: gzip, deflate, sdch
Accept-Language: ru-RU, ru; q=0.8, en-US; q=0.6, en; q=0.4
Accept-Charset: windows-1251, utf-8; q=0.7, *; q=0.3

HTTP/1.1 302 Found
Location: www.google.ru/
Cache-Control: private
```

```
Content-Type: text/html; charset=UTF-8
Set-Cookie:
  PREF=ID=632128af861ebb83:TM=1281420591:LM=1281420591:S=ELyyEhjZ2JNSeGMV;
  expires=Thu, 09-Aug-2012 06:09:51 GMT; path=/; domain=.google.com
Set-Cookie: NID=37=r81qA-
  WGJc3SVSCovGM80YSzQb0LXRe39SHMzqC1X_9reL_
  jYI7q3Y3UKfXAdVtkv8m5fZ11W6XR-
  of4kl0wGAw4LQYK5LTatCv3BOGwcmOc5V_1yhWjo9Hj6q
  bX2acc; expires=Wed, 09-
  Feb-2011 06:09:51 GMT; path=/; domain=.google.
  com; HttpOnly
Date: Tue, 10 Aug 2010 06:09:51 GMT
Server: gws
Content-Length: 218
X-XSS-Protection: 1; mode=block
```

И, затем начинает пинговать гугловый сервис финансовой аналитики:

```
GET /tools/pso/ping?
  as=chrome&brand=&pid=&hl=en&rep=2&rlz=I7:1I7
  GGLL_ru,W1:1W1GGLL_ru,T4:1T4GGLL_ru,C1:1C1
  enRU392RU392,C2:1C2_____enRU392,R7:1R7GGLL_
  ru&dcc=T
HTTP/1.1
Accept: text/*
User-Agent: Mozilla/4.0 (compatible; Win32)
Host: clients1.google.com

HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Date: Tue, 10 Aug 2010 06:10:07 GMT
Expires: Tue, 10 Aug 2010 06:10:07 GMT
Cache-Control: private, max-age=0 ▶
```

```
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Server: GSE
Transfer-Encoding: chunked

rlzT4: 1T4GGLL_ru
rlzR7: 1R7GGLL_ru
rlzW1: 1W1GGLL_ru
rlzC1: 1C1_____enRU392RU392
rlzC2: 1C2_____enRU392
rlzI7: 1I7GGLL_ru
dcc: T
set_dcc:
  T4:1T4GGLL_ru,R7:1R7GGLL_ru,W1:1W1GGLL_ru,C1:1C1_____enRU392RU392,C2:1C2_____enRU392,I7:1I7GGLL_ru
events:
stateful-events:
crc32: 7c33310f
```

В том, что это финансовый сервер, а не какой-нибудь супер-мега-шибоко закодированный полезный сервис можно очень легко убедиться, заглянув в исходники библиотеки RLZ, любезно опубликованной самим Google. Ниже я привожу наиболее интересный нам кусок кода:

```
//
// Ping information.
//

// rep=2: includes the new stateful events.
const char kProtocolCgiArgument[] = «rep=2»;

const char kEventsCgiVariable[] = «events»;
const char kStatefulEventsCgiVariable[] = «stateful-events»;
const char kEventsCgiSeparator = ',';

const char kRlzCgiVariable[] = «rlz»;
const char kRlzCgiSeparator[] = «,»;
const char kRlzCgiIndicator[] = «:»;

const char kProductSignatureCgiVariable[] =
```

```
«as»;

const char kProductBrandCgiVariable[] = «brand»;

const char kProductLanguageCgiVariable[] = «hl»;

const char kProductIdCgiVariable[] = «pid»;

const char kDccCgiVariable[] = «dcc»;
const char kRlsCgiVariable[] = «rls»;
const char kMachineIdCgiVariable[] = «id»;
const char kSetDccResponseVariable[] = «set_dcc»;

//
// Financial server information.
//

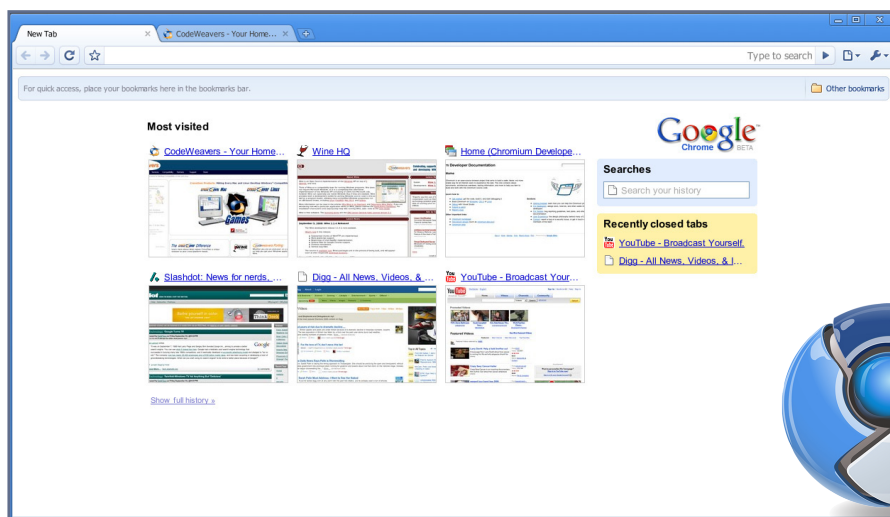
const char kFinancialPingPath[] = «/tools/psop/ping»;
const char kFinancialServer[] = «clients1.google.com»;
const char kFinancialPingType[] = «GET»;

const int kFinancialPort = 80;

// Ping times in 100-nanosecond intervals.
const int64 kEventsPingInterval = 24LL * 3600LL * 100000000LL; // 1 day
const int64 kNoEventsPingInterval = kEventsPingInterval * 7LL; // 1 week

const char kFinancialPingUserAgent[] = «Mozilla/4.0 (compatible; Win32)»;
const char* kFinancialPingResponseObjects[] = { «text/*», NULL };
```

Причем, напомню, что сам гугль утверждает, что данный код используется ТОЛЬКО для маркировки брендированных скачек и его нету даже в Chrome, скачанном с оффсайта, не говоря уж о «в склянь свободном» белом и пушистом Chromium. Обсуждение данной проблемы я поднял в официальной гуглгруппе Chromium-discuss, где получено подтверждение аномальности данного поведения. Интересно, гугл снова «по ошибке» решил подсобрать побольше данных о пользователях? ■





# Клавиатура с подсветкой

Автор

Дмитрий Игнатьюк  
Программист  
г. Луцк, Украина  
[www.Dm.com.ua](http://www.Dm.com.ua)

Я работаю программистом и как многие программисты подрабатываю/дорабатываю на дому ночью (а когда же еще). Но проблема в том, что мне не нравится работать без света (общаться в аське или играть в игры – это совсем другое), а включить свет не могу по той причине, что жена ложится спать.

В инете есть множество статей о том как сделать подсветку для клавиатуры, но везде нужно портить либо клавиатуру, либо стол, а мне это не улыбается. И вот наткнувшись на статью о светодиодной лампе для мелких работ, питающуюся от батарейки-таблетки у меня появилась идея сделать мобильную подсветку для клавиатуры, которая будет питаться от той же таблетки.

## Что нам нужно

Во-первых, конечно же желание. Возможно, оно будет вызвано вдохновением что-то сотворить, а возможно некоторой усталостью писать программы в темноте, но все же без него никак.

Далее детали: 2 светодиода (лучше белых), 2 батарейки-таблетки на 3В, 2 канцелярские прищепки (купить можно в любом магазине канцтоваров и я так понял, что купил самые большие; в любом случае при покупке представьте, как эти прищепки будут держаться за клавиатуру).



А также кусочек витой пары, паяльник, припой, кусачки, спичечный коробок (или кусок другого тонкого картона) и, конечно же, изолянта (ну как тут без нее).

Учитывая вышеизложенное и проверив правильное местоположение рук, можно смело приступать к изготовлению девайса.

## Беремся за работу

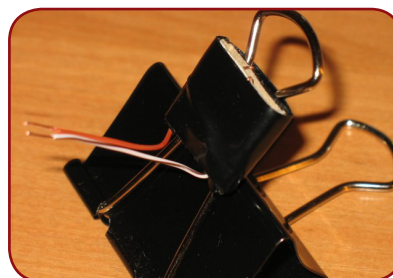
Сначала берем одну из широких сторон спичечного коробка, сгибаем пополам, но не нужно плотно прижимать, внутри должна лежать батарейка, и обрезаем картон так чтобы (со стороны противоположной сгибу) батарейка пряталась примерно на половину как на картинке.



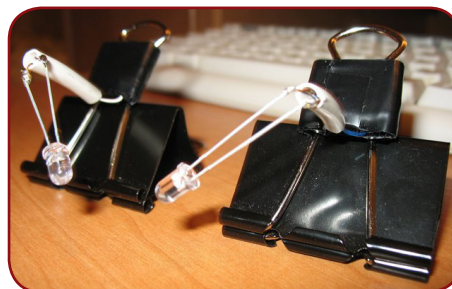
Две другие стороны должны быть чуть длиннее самой таблетки, чтобы надрезав картон по сгибу и завернув, получался карманчик. Далее нужно обмотать карманчик изолентой, при этом желательно, чтобы батарейка была внутри, иначе потом она рискует не поместиться на свое место, входить/выходить она должна туго, но и без молотка.

Потом на сгибе пробиваем две дырки в противоположных сторонах и просовываем туда проводки с витой пары, заранее очищенные от изоляции в том месте, где провод будет находиться в кармане. Загибаем кончики проводов на края кармана так, чтобы батарейка касалась бы к этим проводам разными полюсами.

Теперь нужно обмотать весь карман с проводами изолентой несколько раз для прочности и примотать все той же изолентой карман к прищепке:



Далее припаять к кончикам проводов светодиод. Правда, я еще от нечего делать для красоты одел провода в кожу витой пары. Вторая по аналогии.



## Испытание

Говорят, испытание чего-то сделанного собственноручно стоит того, чтобы сделать это что-то собственноручно (ну и конечно нельзя не учитывать гордость за самого себя).

Ну что я могу сказать. Мне понравилось. Удобно, в глаза не бьет, освещение достаточное. В любой момент можно снять прищепку с клавиатуры и выключить устройство, достав батарейку. ■





# Обзор медиаплеера PowerZest HD-500

Автор

Сергей Кузнецов  
г. Балашиха  
www.kuuzya.ru



В последнее время на рынке появился спрос на медиа-центры (медиаплееры) – устройства, которые делают из телевизора DVD, MP3-плеер, радио, интернет-браузер и т. п., причем есть они и с жестким диском и без. Расскажу вам еще об одном представителе этого класса от фирмы PowerZest, который обладает многими «вкусностями», но, к сожалению, без встроенного HDD.

Начнем как обычно с внешних прелестей и закончим на возможностях.



Выпускается HD-500 вот в такой коробке, на которой помимо изображения устройства перечислены все его достоинства: это и прослушивание радио Shoutcast, и просмотр YouTube и многое другое.



Внутри коробки помимо самого девайса находятся:

- Пульт ДУ
- Батарейки для пульта
- HDMI кабель
- AV кабель
- Адаптер для зарядки
- Ethernet-провод

Единственное, что хочется к этому добавить, устройством можно управлять только с помощью пульта, поэтому его потеря или поломка принесет много дискомфорта. На самом девайсе есть только одна кнопка On/Off.



На нем есть много кнопок для доступа к специальным возможностям напрямую, а не через меню. А также буквенно-цифровая клавиатура, которая пригодится для web-серфинга, о котором чуть-чуть попозже.

Также, конечно, в комплекте идет достаточно толстая инструкция, которую я советую почитать, так как с некоторыми функциями сложно самому разобраться.

## Внешний вид

Устройство черное глянцевое, на передней стороне есть только ИК-приемник для пульта. На задней же панели расположен ряд разъемов: тут и USB, и HDMI, и Ethernet, и AV, а также зеленая кнопка включения-выключения. Девайс выглядит очень красивым, и я думаю, впишется в дизайн любого интерьера, ведь черный универсальный цвет. Единственный недостаток корпуса – это то, что он глянцевый и оставляет на себе все отпечатки.



Снизу кроме резиновых ножек и наклейки ничего нет. Это и правильно, ведь на стену девайс не повесишь, так как неудобно будет управлять, а вот устойчивость должна присутствовать.

Приведу еще некоторые технические спецификации:

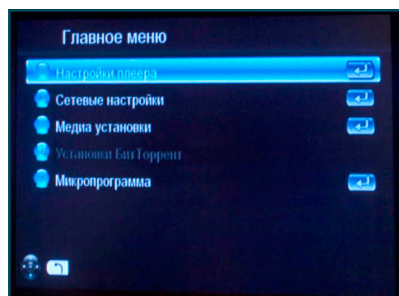
- Выходы аудио стерео, аудио оптический, видео композитный, HDMI 1.3
- Интерфейсы USB Type A x2, Ethernet, eSATA Host
- Скорость Ethernet 10/100 Мбит/с
- Поддержка онлайн-сервисов видео, фото, радио-сервисы
- Форматы файлов MKV, AVI, WMV, ISO, IFO, VOB, MOV, ASF, TP, TS, M2TS, MP4, MPG
- Кодеки MPEG1, MPEG2, MPEG4, H.264, WMV9, VC1
- Аудиофайлы MP3, WMA, AAC, Ogg, PCM, FLAC, AC3, DTS ▶

- Графические файлы JPEG, GIF, BMP, PNG, TIFF
- Поддержка субтитров srt, ssa, idx, sub, smi

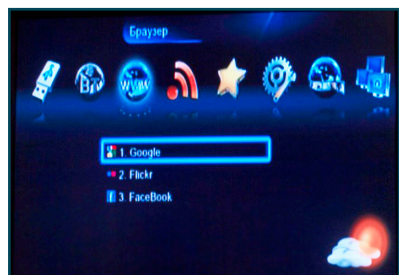
Все эти форматы проигрываются очень хорошо и быстро. Даже HD 1080p идет без всяких тормозов. Субтитры подхватываются из папки с фильмом, в общем все, как и на компьютере.

## Взаимодействие с телевизором

Как уже говорилось, включить устройство можно двумя способами (кнопкой на корпусе и пультом). После включения вы увидите заставку, а за ней вы попадаете в меню настройки, где можно сделать все необходимые приготовления (яркость, громкость, тип соединения и т. п.) для комфортной работы с устройством.



Одной из интересных возможностей этого плеера является возможность путешествовать по просторам интернета прямо через телевизор. Первое – это обычные сайты, изначально на выбор дается Google, Facebook и Flickr. Но, конечно, вы можете посещать и кастомные сайты.



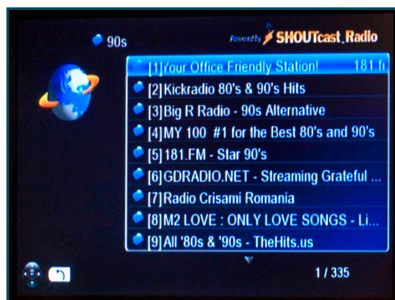
Мне было очень интересно, как это сидеть на Facebook через телевизор. Достаточно забавно. Конечно, набирать текст с помощью цифровой клавиатуры – это как на телефоне, но вполне приемлемо.



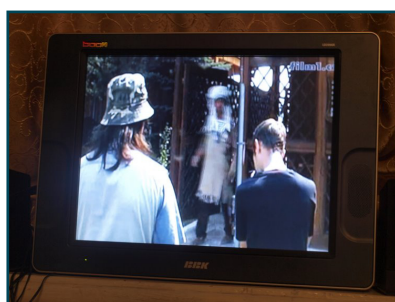
Кстати, если заметили, то плеер автоматически скачивает информацию о погоде в вашем городе (на основе IP) и выводит ее внизу. А вот так выглядит проигрывание видео с Youtube. Как видите, видео встраивается в стандартный плеер.



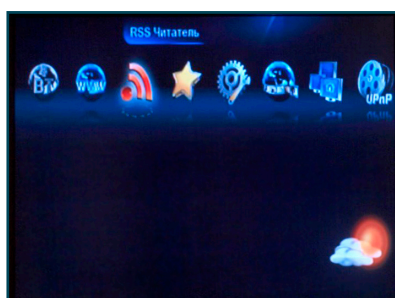
Также, вы можете слушать интернет-радиостанции, которые предоставляет пожалуй самый популярный радиохост Shoutcast.



Несомненно, можно подключиться и к интернет-телевидению. Вот это, например, просмотр бельгийского канала.



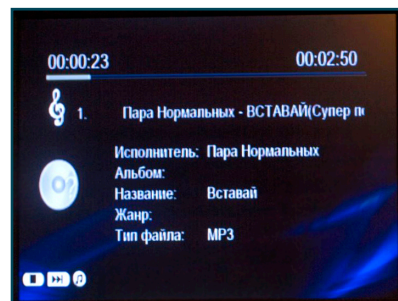
А если вы настроите RSS каналы, то сможете еще и читать новости через телевизор.



Не зря сделан USB порт, ведь вставив флешку или любой другой накопи-

тель (камеру, фотоаппарат, внешний HDD), можно смотреть фильмы, фотографии и слушать музыку.

Довольно удобно, причем музыка продолжает играть и в фоновом режиме, то есть можно смотреть фото и слушать музыку одновременно.



В дополнение хочу сказать, что в HD-500 есть возможность торрентов, соединения с сетевыми устройствами и проигрывания с них медиафайлов. Ну и разработчики поддались девизу «Everything better with Bluetooth», поэтому с плеера можно соединиться с ВТ устройством (телефоном) и прослушать с него музыку. Правда, у меня это сделать не получилось.

Вот такие штуки предоставляет эта маленькая черная коробочка. Цена на устройство составляет около 5 000 р, что вполне недурно для девайса такого качества.

## Подведение итогов

В целом, я считаю, что в полку недорогих медиаплееров прибыло, причем это хорошее пополнение. Недорогой, мощный, функциональный и просто красивый. Конечно, интерфейс немного уменьшил мои восторги, хотя наверно это больше зависит от самого телевизора, но фильмы проигрываются в хорошем качестве и без зернистости. А в остальном – это мечта. Обилие функций, встроенный браузер, Youtube, Shoutcast, связь по Ethernet и Bluetooth. За такую цену это вполне приемлемо. А что еще надо, если вы просто хотите сделать из телевизора мультимедиа-плеер? ■

|   |                            |
|---|----------------------------|
| + | Дизайн                     |
|   | Обилие возможностей        |
|   | Браузер                    |
|   | HDMI и AV                  |
| - | Цена                       |
|   | Отсутствие HDD и Wi-Fi     |
|   | Не очень удобный интерфейс |



# FastVPS.ru

VPS от 129 рублей

- 100 MB RAM
- 300 MHz CPU
- 6 GB HDD

Dedicated от  
2199 рублей

- Intel® Core™ i7-920  
Quad-Core incl
- 8 GB DDR3 RAM
- 2 x 750 GB HDD



- один из лучших  
дата-центров Германии
- служба поддержки  
на русском языке
- уникальная система  
восстановления
- удаленный *reboot/reinstall*
- установка сервера  
в течение 24 часов