

Даниил Шидловский

Электронный управленец

Проблема поиска постоянного или дополнительного заработка с помощью разработки свободного программного обеспечения легко решается при наличии опыта, знаний и умения небанально мыслить.

Радость от совместной жизни с любимой девушкой длилась недолго. Остро встал финансовый вопрос. А если учесть, что я ленивый, к тому же еще и разгильдяй, не способный работать в команде, то проблема представлялась очень даже серьезной. Между прочим, мое трудоустройство на постоянную (даже высокооплачиваемую) работу не устраивает и мою девушку тоже. Кроме денег она требует постоянного внимания.

В раздумьях о заработке я отправился на книжный рынок и купил издание из серии «Бедный папа, богатый папа». Одна мысль меня очень заинтересовала: чтобы стать обеспеченным — лучше получать \$100 в месяц и почти ничего не делать, чем \$5000, вкалывая от зари до зари. Потому что \$100 пассивного дохода позволят найти следующую сотню. А потом еще. И еще.

Подготовка

Дальше я поведаю вам о том, как моей мечте удалось воплотиться в реальность. Мы вместе пройдем весь путь от поисков, робких экспериментов и закончим решением тех проблем, с которыми я столкнулся при коммерческой эксплуатации продукта, получившего название «Докотека».

Итак, начнем с конца. Нужны деньги. Не очень большие, но систематически. У кого есть деньги? У больших компаний. Имея определенный опыт делового общения, я сразу отказался от этой идеи. Ведь там сплошная политика. А я далеко не Лебедев.

Другой вариант — обычные пользователи. Тоже мимо кассы. Тот, кто настраивал компьютеры частным лицам, меня поймет. Работы очень много, денег — кот наплакал. Остановился на следующем варианте: мои клиенты — владельцы небольшого бизнеса. Зарегистрировав предприятие, я сам легко и непринужденно попал в этот круг. И сразу остро почувствовал необходимость в инструменте управления информацией. И теперь он у меня есть — «Докотека».

Используя личный опыт и несколько замечательных продуктов с открытым исходным кодом, вы сможете запустить прекрасный сервис практически не выходя из дома. Да, большинство клиентов привозят мне деньги прямо домой. Ценят. Вот неполный список необходимых в работе продуктов: Linux, Python, Roundup, PostgreSQL, Mtacs, Freemind.

Что необходимо разработчику?

Внутреннее содержание проекта должно отвечать нескольким критериям. Как оказалось, примерно следующим:

- ▶ очень простое (лень, знаете ли);
- ▶ распространенное;
- ▶ на Python;
- ▶ с открытыми исходными кодами;
- ▶ созданное профессионалами, желательно для себя.

Внешний облик должен иметь веб-интерфейс (но очень быстрый). Проект должен работать на небольшом сервере на основе недорогой машины под управлением, естественно, Linux. И еще обязательно нужно использовать MySQL с возможностью перехода на PostgreSQL.

В ходе поисков оптимального решения я перебрал несколько проектов. Перечислю их, вкратце упомянув плюсы и минусы.

- ▶ Roundup (roundup.sourceforge.net). Собственно, понравилось все: дизайн, документация, исходный код. Стабильное развитие.
- ▶ Zope2 (www.zope.org). Среда с непростой системой безопасности и требовательна к ресурсам. Использую для веб-разработок.
- ▶ Zope3 (www.zope.org). Не потянул. Отладка ZCML (это больше, чем конфигурационный файл) показалась для меня сущим адом.
- ▶ Quixote (www.mems-exchange.com). Хорошо. В связке с SQLAlchemy пишется очень легко. Но захотелось чего-нибудь нового (для меня) и более распространенного.

► Tiny erp (www.tinyerp.org). Сильный продукт. Обнаружил его, когда «Докотека» уже была на полпути к клиенту. Уверен, что еще вернусь к нему.

Остановился на Roundup по той причине, что этот продукт используют профессионалы, к тому же он позволяет менять схему данных прямо на лету. А это для меня самое важное, ведь вначале я не имел представления о том, какие структуры буду задействовать. Кроме того, Roundup автоматически ведет журнал операций, работает с dbm, MySQL, PostgreSQL, SQLite на выбор. Данными можно управлять из командной строки, через Интернет или SMTP.

Задачи, решаемые «Докотекой»

Какие же задачи должна уметь решать система? Руководителю бизнеса от нее требуется следующее:

- централизованное хранилище файлов с доступом через Глобальную сеть;
- отчет о том, кто какую работу выполнил и сколько времени это заняло;
- быстрый доступ к нужному файлу;
- резервное копирование всей информации;
- обмен сообщениями.

Разработка

Ну что, поехали? Для начала я поднял Linux на стареньком Celeron 500. Затем скачал и установил следующие компоненты:

- Python (интерпретатор языка программирования);
- MySQL (движок базы данных);
- MySQLdb (модуль Python для работы с MySQL);
- Roundup (bug tracking).

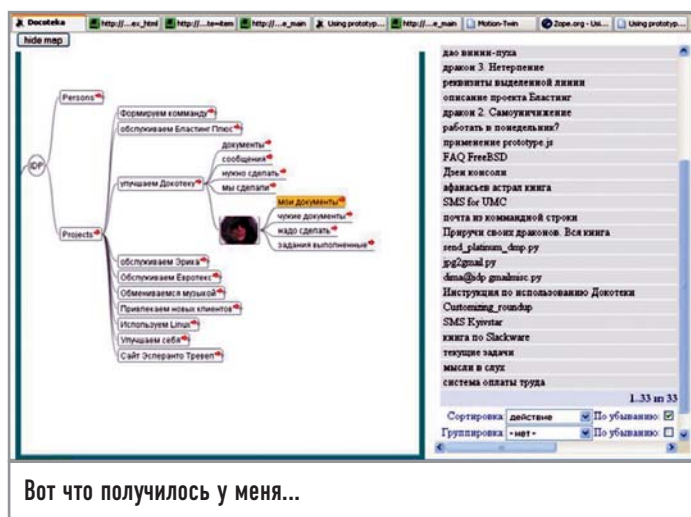
Затем установил администраторский модуль roundup, набрав в командной строке:

```
$ roundup-admin install
```

При установке выбрал шаблон (template) classic и MySQL в качестве backend:

```
$ emacs /trackers/classic/config.ini
```

Здесь я редактирую переменные. Самые важные из них — web, domain, host. Теперь необходимо указать пароль администратора и каталог с трекером:



```
$ roundup-admin initialise
```

```
$ roundup-server classic=/home/myhome/trackers/classic
```

И можно запускать roundup.

А затем открывать любимый Firefox на <http://localhost:8080/classic>.

Получился баг-трекинг, полностью готовый к работе. Уже локализованный! Очень хорошо. Но этого мало. Распечатал [customising.html](#), [client.py](#) и поехал на неделю на дачу (компьютер не брал). Изучал до просветления. Стал думать над интерфейсом. Не нашел лучшего решения, чем дерево и список. Roundup замечательно работает со списками из коробки. А вот с деревом не все так просто. Может, есть варианты получше? Случайно наткнулся на продвинутую модель дерева — MindMap. Дальше MindMap я буду называть ментальной картой; подробнее о ней можно узнать на сайте www.mindmap.ru

Нашел свободный продукт для работы с ментальными картами — Freemind (<http://freemind.sourceforge.net>). На скриншоте вы можете видеть пример его работы.

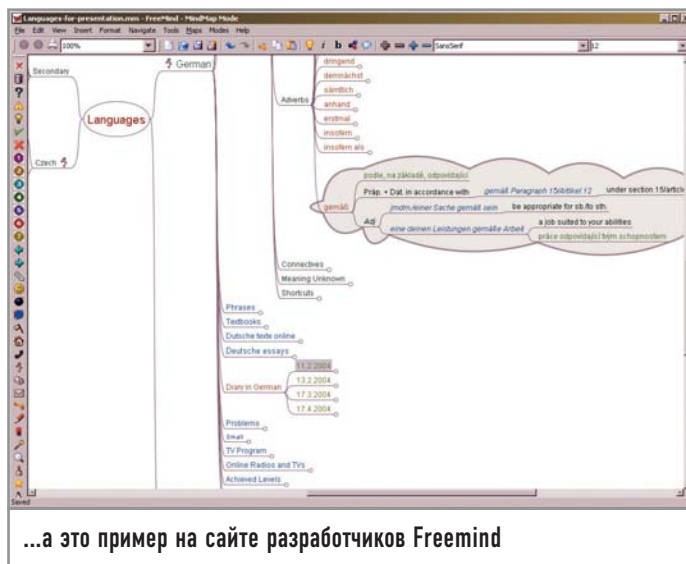
Потом написал скрипт на Python, который генерирует XML, на основе которого Freemind строит ментальную карту.

Жаль только, что для отображения ментальной карты в браузере приходится использовать java plugin. И тут удача улыбнулась мне. По адресу <http://cvs.sourceforge.net/viewcvs.py/freemind/flash> я нашел исходники к SWF-файлу (Macromedia Flash, если кто не знает), который отлично отображает ментальную карту в Flash plugin для браузера.

И что самое главное, этот visorFreemind.swf создан с помощью свободного компилятора (подробнее о нем вы можете узнать на сайте www.mtasc.org). Какая чудесная находка!

Видение того, каким должен быть интерфейс, пришло ко мне во сне: слева — ментальная карта, причем у каждого пользователя своя; справа — фрейм для отображения данных. При нажатии на узел слева или справа открывается форма для ввода данных или список.

Проснувшись, сразу же сел за компьютер. Помедитировал над исходниками visorFreemind.swf и внес два небольших изменения. В файле Main.as убрал строку `System.useCodepage = true`, а в Node.as во всех вызовах `getUrl` добавил еще один параметр —



right_frame, чтобы ссылки открывались в правом фрейме (да, да, я помню, что фреймы использовать не рекомендуется, но ничего лучше конструкции с тегом `<iframe>` не придумал).

Компилирую с учетом того, что исходники лежат в каталоге `/home/myhome/freeMindFlash/visorFreeMind`, следующей строкой:

```
$ mtask -cp ./freeMindFlash/visorFreeMind/Main.as -swf
visorFreemind.swf -header 600:600:24 -separate
```

Несколько слов о том, как будет храниться информация. В Roundup схема данных представлена множеством классов, размещенных в файле `schema.py`. Вот выдержка из моего `schema.py`, описывающая класс `Document`:

```
doc = Class (db, "doc", title=String(), file=Link("file"),
state=Link("docstate"), doctype=Link("doctype"),
project=Link("project"), interests=Link("user"),
versions=Multilink("file"), messages=Multilink("msg"))
```

Здесь, `doc`, `file`, `docstate`, `doctype`, `project`, `user`, `msg` — наименования классов. `String`, `Number`, `Date`, `Interval`, `Number` — свойства экземпляров. `Link`, `Multilink` — связи между экземплярами классов. Если что-то не ясно, то документ `customising.html` — ваш лучший друг!

Стоит заметить, что добавить или удалить класс, свойство или связь я могу просто посредством редактирования файла `schema.py`. Таблицы в базе данных изменит за меня Roundup. Здорово, не так ли? Но нужно соблюдать известную осторожность.

Запросы — одна из удивительно приятных и мощных особенностей Roundup. Вот как будет выглядеть URL, для того чтобы вывести список документов созданных мной. Этот URL привязан к узлу «Мои документы», как показано на скриншоте:

```
http://mytracker:8080/docotheka/doc?@template=index&@columns=
title,file&@filter=creator&creator=3&@pagesize=50&@sort=-activity
```

Необходимые пояснения:

- ▶ `doc?@template=index` — используется шаблон `doc.index.html`;
- ▶ `@filter=creator&creator=3` — будут выведены документы, где свойство `creator = 3`;
- ▶ `@columns=title,file` — будет две колонки — заголовок и ссылка на файл;
- ▶ `@pagesize=50` — 50 записей на странице;
- ▶ `@sort=-activity` — документы, изменявшиеся последними, будут расположены наверху.

Дополнительная информация

Легкий путь к AJAX

С содроганием вспоминаю, как сразу после изучения Perl столкнулся с необходимостью писать на JavaScript. Кошмар. Поэтому всегда старался как можно больше всего делать на стороне сервера. Но вот увидел, что эти парни из Google сделали из почтового клиента, и тогда прозрел. Тоже захо-

телось приобщиться к модному слову AJAX. Трудностей было много, пока не наткнулся на замечательную статью одного специалиста, где он рассказывает о том, как использовать `prototype.js`. Теперь все налажилось. Рекомендую — www.sergio-pereira.com/articles/prototype.js.html.

А теперь поговорим о шаблонах, благодаря которым мы вводим формы для ввода данных и показываем списки. Кстати, практика показала, что вводить данные через формы практически не приходится. Нужно максимально интегрироваться в существующий документооборот. Но об этом немного позже.

Шаблоны представлены файлами в папке `html` и написаны с использованием ZPT (`Zope Page Template`). Изучить этот как бы язык оказалось для меня самым сложным. Но мотивировав себя тем, что это решение для предприятий, которое широко используется не только в Zope (а и в Roundup, например), и вооружившись отличной документацией (www.zope.org/Documentation/Books/ZopeBook/2_6Edition/ZPT.stx), осилил это дело. И оно пошло лучше, когда я осознал, что, используя ZPT, нужно придерживаться простого правила: каждый тег может быть рассмотрен отдельно; никаких `else`.

Дальнейшую радость открытий и удовольствие от постижения исходного кода оставляю вам. Подробности и документация находятся на <http://roundup.sourceforge.net>, а также на моей страничке www.idp.com.ua/docotheka.

| Внедрение |

После того как разработка приложения вошла в свою колею, настало время вернуться к основной задаче: как на этом заработать? Составил небольшой список того, что может программный комплекс, и пригласил потенциального клиента на кружку пива. Далее расскажу о выводах, которые мне пришлось сделать.

| Что вызывает интерес |

- ▶ Дистанционный контроль над деятельностью сотрудников.
- ▶ Скорость доступа к информации (надо показывать).
- ▶ Низкие расходы.
- ▶ Резервное копирование данных на Gmail.
- ▶ Трудовая дисциплина.

Интерфейс и особенно его скорость клиенту понравились. Все-таки технология Flash имеет в этой области неоспоримые преимущества. Еженедельный отчет о деятельности сотрудника стал своеобразной «бомбой». Выяснилось, что клиент может платить \$150 в месяц. Я подумал и решил понизить цену, чтобы оплата включала предоплату и происходила без задержек: €100 в месяц. Ударили по рукам.

Приехал, установил. В качестве движка базы данных использовал PostgreSQL. Первый день работы принес несколько проблем, среди которых — небольшой баг в генерации карты и непонимание пользователями принципов работы. Пришлось установить VNC (www.tightvnc.com) на каждый компьютер клиента. После этого дело пошло гораздо быстрее.

Главная проблема возникла через шесть недель. «Докотекой» перестали пользоваться. Сотрудники предпочитали по старинке закидывать файлами свои рабочие столы. Не стал с ними бороться, так как и сам такой же ленивый. Решением стал описанный в прошлом номере небольшой скрипт, который в конце дня обходит рабочие каталоги пользователя и все измененные файлы загружает в «Докотеку». Дело налажилось. А за одним клиентом потянулся другой... Теперь обкатываем учет платежей. В следующий раз отчитаюсь. |