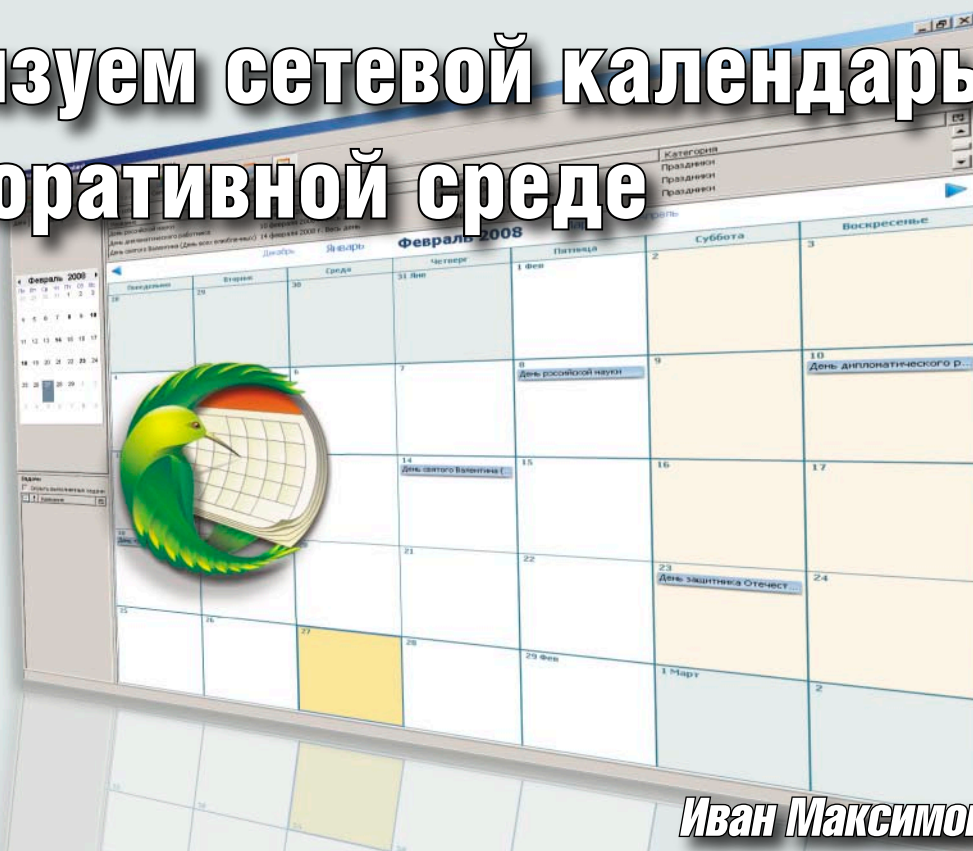


Организуем сетевой календарь в корпоративной среде



Как организовать коллективный доступ к корпоративному ежедневнику? Возможно ли при этом настроить аутентификацию пользователей и обеспечить безопасность? Среди разнообразия программных продуктов найдем тот, который понравится пользователям и предоставит необходимый функционал.

«**К**ак расшарить календарик в Outlook?» – к подобным вопросам со временем привыкаешь, но все же они вводят в ступор. Задают их чаще всего наши друзья, партнеры, клиенты, поэтому приходится объяснять и рассказывать на пальцах, что ту или иную задачу не всегда можно решить двумя кликами мыши. Чаще всего это лишь начало – раскрывая детали и постепенно реализуя тот или иной проект, понимаешь, что он начинает обрастать всевозможными доработками и расширениями. Но не будем ходить вокруг да около, давайте «расшарим календарик».

Очертим круг задачи: необходимо настроить простой, элегантный, русскоязычный, интуитивно понятный сетевой календарь, аналогичный или на базе ежедневника Microsoft Outlook. Под сетевым понимается то, что доступ должен быть как в пределах локальной сети предприятия, так и из лю-

бой точки мира через Интернет. Календарь должен быть кроссплатформенным и безопасным в использовании. Дополнительное, но необязательное требование – синхронизация через ActiveSync со всевозможными КПК, смартфонами, работающими под управлением Microsoft Windows Mobile. И последнее – вопрос о лицензионности программного продукта становится в России все более и более актуальным, связано это с тем, что 1 января 2008 года вступил в силу закон об авторском праве, укрепляющий позиции производителей коммерческого программного обеспечения (см. Гражданский кодекс Российской Федерации от 18 декабря 2006 г. №230-ФЗ Часть четвертая [1]), поэтому продукт по возможности должен распространяться по свободной лицензии.

Выбор ПО

Microsoft Outlook, в связи с требованием кроссплатформенности, не гово-

ря уже о цене, нам не подойдет. Рассматривая статьи Сергея Яремчука в журнале «Системный Администратор» за прошедший 2007 год о различных средствах групповой работы, работающих через веб-интерфейс, можно выделить несколько весьма интересных решений. Данные комплексы предоставляют широкий функционал, в чем-то даже намного больший, чем требуется, но не все они поддерживают русский язык, далеко не все имеют возможность синхронизации с мобильными устройствами, но главное, при тестовом внедрении phpGroupWare и eGroupWare (именно этот путь и был выбран в начале реализации проекта) от пользователей можно было услышать мнение, сыгравшее немаловажную роль – «топорный дизайн». Все-таки нужно признать, что HTML- и JPG-файлы не самые лучшие элементы дизайна.

По ходу экспериментов с различными средствами для организации груп-

повой работы все же было найдено программное решение для реализации поставленной задачи. Было ясно, что для удовлетворения дизайнерских прихотей пользователей необходима полноценная клиентская программа, аналогичная Microsoft Outlook; таковыми являются: Evolution, созданная в недрах компании Novell, Kontakt, входящая в состав KDE, и, конечно же, Mozilla Sunbird, последняя доступна еще и как расширение к почтовому клиенту Thunderbird, только под другим названием – Lightning.

Наибольший интерес у пользователей вызывала Sunbird. К плюсам данной программы можно отнести:

- кроссплатформенность – продукция Mozilla прекрасно работает как в семействе ОС Windows, различных *nix-системах, так и в Mac OS;
- русскоязычный интерфейс;
- свободную лицензию.

Стоит упомянуть и о том, что продукция этой компании прошла проверку временем: в январе 2008 года Mozilla праздновала юбилей – 10 лет. Определившись с клиентской программой, рассмотрим подробно настройку нашего комплекса.

Программное обеспечение и протоколы передачи данных

Каким образом осуществляется работа по сети в Sunbird? Ответ на данный вопрос можно найти на заглавной странице проекта [2], где написано, что, для того чтобы календари были доступны нескольким пользователям, необходимо использовать «WebDAV server». Что это за сервер и сервер ли вообще? Обратившись к базе данных rfc [3], можно найти документ за номером 2518 и его замену, опубликованную в июле 2007 года – rfc4918. Эти документы описывают базовую версию протокола высокого уровня WebDAV, аббревиатура которого расшифровывается как Web-based Distributed Authoring and Versioning. Данный протокол работает поверх стандартного HTTP и дополняет его специфичными командами, в первую очередь касающимися совместной работы по сети над файлами. Но не будем сильно акцентировать внимание на самом протоколе: пройдя по ссылке, указанной выше, можно найти порядка 10 документов, описывающих данный протокол.

Итак, разобравшись с протоколом, определимся, каким образом будет задействован WebDAV. На форуме mozilla-russia.org не раз уже поднимался вопрос о том, как лучше это сделать: кто-то предлагает использовать CalDAV, кто-то использует кроссплатформенный сервер COSMO, есть и те, кто использует модуль mod_dav к веб-серверу Apache. Так как кроме самой реализации протокола WebDAV необходимо обеспечить аутентификацию пользователей и безопасность передачи данных, воспользуемся «старым добрым» веб-сервером Apache, настроив его в связке: Apache2 + mod_dav + mod_ssl + mod_auth.

Тут стоит немного задержать внимание: дело в том, что в разных версиях данного веб-сервера модуль WebDAV ре-

Сетевой календарь – быстрый способ

Самый простой и быстрый способ организовать работу Sunbird по сети – это использовать протокол SMB для передачи данных на серверах, а говоря проще – обыкновенную открытую по сети папку в ОС Windows. После установки календаря перейдем в меню «Файл → Новый календарь → В сети →

iCalendar (ICS)». В диалоговом окне выбора расположения файла с заметками введем строку вида file:///FileServer/Share/Calendar.ics. Но данный способ организации сетевого доступа можно порекомендовать лишь в домашних условиях, где используется 2-3 компьютера, либо в экстренных случаях, с доступом только для чтения, при нарушении работы основного календаря.

ализован по-разному, поэтому конфигурационные файлы в Apache 1.3/2.0/2.2 будут разными (файлы будут незначительно отличаться), более подробно об этих различиях можно прочитать на русском языке здесь: <http://apache.dev.ru>.

Последнюю деталь – синхронизацию с мобильными устройствами – возможно организовать посредством BirdieSync или FinchSync.

Настройка сервера

Сервер, на котором будет осуществлена настройка, работает под управлением ОС Debian Linux 4. Несмотря на эту небольшую специфичность, большая часть действий, выполняемая на сервере, будет справедлива и для других *nix-систем, кроме, конечно же, базовых, связанных с установкой пакетов.

Воспользовавшись стандартной утилитой управления пакетами aptitude, устанавливаем веб-сервер и модули к нему:

```
aptitude install apache2 libapache-mod-dav davfs2
```

Далее ставим все необходимое для поддержки шифрования, OpenSSL и нужные библиотеки:

```
aptitude install openssl libssl-dev
```

mod_auth, mod_ssl и другие базовые модули для веб-сервера Apache2.0 будут установлены автоматически при исполнении указанных выше команд.

Хотя порядок настройки может быть иным, в первую очередь займемся SSL. Как известно, безопасное соединение обеспечивается по протоколу HTTPS на стандартном 443 порту TCP. Работает это таким образом: клиент подключается к серверу, модуль SSL устанавливает алгоритмы шифрования и передает ему открытый ключ в зашифрованном сертификате. После установки шифрования идет безопасная передача данных. Сами сертификаты можно разделить на 3 группы:

- **Самоподписные.** Открытый, закрытый ключи и сертификат создается самостоятельно.
- **Сертификат, подписанный доверенным СА.** Сертификат подписывается доверенным центром сертификации.
- **Сертификат, подписанный локальным СА.** Используется в больших локальных сетях с собственным центром сертификации.

Стоит выбирать между первым и вторым вариантами.

Первый путь имеет один большой недостаток: чтобы клиентские программы (например, браузеры) пользователей могли без проблем идентифицировать сервер и его сертификат, необходимо установить данный сертификат в каждую клиентскую программу. В рассматриваемом случае используем данный метод только для тестирования комплекса, хотя для осуществления безопасной передачи данных в небольших сетях (да и средних тоже, все зависит от поставленных задач) самоподписные ключи вполне можно применять для работы.

Самоподписный сертификат создается так:

```
openssl req -x509 -nodes -newkey rsa:1024 -keyout ./etc/apache2/server.key -out ./etc/apache2/server.crt
```

что означает:

- **req** – данная команда отвечает за создание сертификата с указанными параметрами;
- **-x509** – создаем новый сертификат формата x509;
- **-nodes** – ключ не защищен парольной фразой;
- **-newkey rsa:1024** – закрытый ключ будет создан, используя алгоритм rsa с длиной ключа 1024 бит;
- **-keyout** – путь сохранения файла закрытого ключа сервера;
- **-out** – путь сохранения файла сертификата.

Второй путь считается более правильным, так как наш сертификат будет подписан авторитетным центром сертификации, но где же найти эти авторитетные центры? В продукции Mozilla, а конкретно Firefox, их можно найти, пройдя по пунктам «Инструменты → Настройки → Дополнительно → Шифрование → Просмотр сертификатов → Центры сертификации». Выбирать стоит кого-то пораспространенней, подешевле или даже бесплатных (последние существуют). В нашем конкретном случае достаточно выбрать кого-то из перечисленных в Firefox, так как продукцию компании Microsoft мы не будем использовать. Не стану рекламировать кого-то конкретно, могу лишь отметить что: COMODO CA Limited выдает ключи на 90 дней, но после регистрации высылает ключ, зарегистрированный на EssentialSSL, которого нет в списке доверенных центров Firefox (который, правда, есть в Internet Explorer), поэтому от него пришлось сразу отказаться. Достаточно широко распространены www.thawte.com бесплатно раздает ключи сроком на 21 день, но он отказывается регистрировать локальные ресурсы и IP-адреса, т.е. он будет проверять доменное имя на существование в Интернете.

Итак, на практике процесс генерации и подписи сертификата происходит так. Создадим закрытый ключ и сертификат, требующий подписи, известный так же как Certificate Signing Request (CSR):

```
openssl req -new -nodes -keyout ./etc/ssl/private/myserver.key -out ./etc/ssl/private/respotse.csr
```

```
./etc/ssl/private/myserver.key -out ./etc/ssl/private/respotse.csr
Generating a 1024 bit RSA private key
.....+++++
writing new private key to './etc/ssl/private/myserver.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:RU
State or Province Name (full name) [Some-State]:Russia
Locality Name (eg, city) []:St.Petersburg
Organization Name (eg, company) [Internet Widgits Pty Ltd]:XXX Ltd
Organizational Unit Name (eg, section) []:XXX Ltd
Common Name (eg, YOUR name) []:192.168.0.1
Email Address []:ivan_maksimov@inbox.ru
```

Отличия от команды первого примера незначительные, замечу лишь, что по умолчанию закрытый ключ создается по алгоритму rsa длиной 1024 бита.

Полученный сертификат будет иметь подобный вид:

```
# cat ./etc/ssl/private/respotse.csr
```

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBzjCCATcCAQAwY0xCZAJBgNVBAYTA1JVMQ8wDQYDVQQIEwZSc0XNzaWEeFjAU
BgNVBAcTDVNOI1BldGVyc2JlcmcxZjAMBgNVBAoTbUtzYW5mMQ4wDAYDVQQLEwVl
...
-----END CERTIFICATE REQUEST-----
```

Именно этот файл и необходимо отправить авторизованному центру. Чаще всего существует форма на сайте, в которую необходимо вставить тело сертификата. Обратите внимание, что при генерации новой пары ключей некоторые центры сертификации просят указывать специализированные параметры ключа, например, упомянутый уже COMODO CA Limited просит в поле Common Name указывать: либо полное доменное имя ресурса, либо название ресурса, либо внутренний IP-адрес портала. В приведенном примере было введено 192.168.0.1 – адрес тестируемого сервера. Данные при регистрации следует указывать верные, т.к. после прохождения всех проверок на ваш адрес будет выслан сертификат, содержащий в себе указанные при регистрации данные. Получив подписанный сертификат, поместим его в каталог с ключом сервера, в нашем случае это папка /etc/ssl/private/.

Выполнив необходимые настройки с OpenSSL, перейдем к веб-серверу.

Для начала создадим все необходимые файлы, каталоги и загрузим тестовый календарь.

Создадим каталог для будущих файлов:

```
# mkdir /mnt/disk/calendar/data/
```

Для обеспечения корректного многопользовательского доступа по протоколу WebDAV необходимо создать пустой файл (не содержащий ни одного символа) базы данных блокировок DavLockDB:

```
# cat "" > /mnt/disk/calendar/DavLock
```

Для проведения тестов загрузим с помощью утилиты wget календарь русских праздников с сайта www.mozilla-russia.org:

```
# wget http://www.mozilla-russia.org/projects/calendar/ ./data/RussianHolidays.ics
```

Для организации простейшей аутентификации пользователей средствами веб-сервера создам файл .htpasswd

в каталоге с файлами, доступ к которому необходимо ограничить. По соображениям безопасности веб-сервер Apache запрещает доступ извне ко всем файлам, начинающимся с .ht, так что никто из клиентов не сможет скачать или просто просмотреть этот файл. Нужно заметить, что ни DavLock, ни .htpasswd не обязаны размещаться в дереве веб-сайта — их вполне можно разместить в другом месте системы, в приведенном примере это сделано из соображений удобства (все, что относится к WebDAV каталогу, хранится в одном месте). Сам файл создается утилитой htpasswd, входящей в состав apache2.

```
# htpasswd -c /mnt/disk/calendar/data/.htpasswd admin
```

После выполнения команды будет предложено ввести пароль для нового пользователя и повторить его. В будущем, для обновления данных, ключ -c (Create a new file) не нужен. Пожалуй, стоит уточнить, что подобная организация проверки подлинности пользователей снижает производительность веб-сервера Apache. Если планируется использовать календарь в большой организации, стоит подумать об организации аутентификации с помощью SQL-баз данных.

И последняя команда, связанная с правами доступа:

```
# chown -R www-data:www-data /mnt/disk/calendar/
```

Команда сменит рекурсивно владельца каталога на «www-data» (обычно от имени этого пользователя работает веб-сервер Apache).

Перейдем к конфигурационному файлу. Первое, что нужно сделать, это подключить к Apache2 необходимые для работы модули, поместив в apache2.conf (у меня он расположен в папке /etc/apache2/) строки:

```
# Модуль, отвечающий за работу протокола WebDAV
# и взаимодействие с хранилищем
LoadModule dav_module /usr/lib/apache2/modules/mod_dav.so
# Модуль, реализующий функциональность хранилища ресурсов
# WebDAV на файловой системе
LoadModule dav_fs_module /usr/lib/apache2/modules/mod_dav_fs.so
# Модуль обеспечивающий шифрование SSL
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
```

Далее добавим в тот же конфигурационный файл строки, определяющие работу WebDAV-каталога:

```
# Слушать 443 порт, стандартный для https-соединений
Listen 443

# Принимаем запросы на виртуальные серверы
# со всех адресов, порт 443
NameVirtualHost *:443

# Разрешаем SSL-кэширование. Используем SHM - shared memory.
# Если не выставлять данную опцию, производительность
# веб-сервера значительно снижается
SSLSessionCache shm:/usr/local/apache2/logs/ssl_cache_shm

# Указываем время в секундах для кэшируемых сессий
SSLSessionCacheTimeout 600

# Открываем контейнер виртуального хоста
<VirtualHost *:443>

# Указываем расположение файла базы данных блокировок
# WebDAV
DavLockDB "/mnt/disk/calendar/DavLock"
```

```
# Ассоциировать для клиентов путь https://HOST/uploads
# с физически расположенной на жестком диске папкой
# /mnt/disk/calendar/data/
Alias /uploads "/mnt/disk/calendar/data"

# Включаем поддержку SSL/TLS на виртуальном хосту
SSLEngine on

# Указываем расположение файла сертификата
SSLCertificateFile /etc/ssl/private/192_168_0_1.crt

# Место расположения закрытого ключа сервера
SSLCertificateKeyFile /etc/ssl/private/myserver.key

# Директива ограничивает диапазон команд в пределах
# веб-адресов
<Location /uploads>
# Включаем поддержку WebDAV
DAV On
# Аутентификация базовая
AuthType Basic
# Описание аутентификации
AuthName "Введите пароль для доступа к календарю"
# Расположение файла с базой пользователей
AuthUserFile /mnt/disk/calendar/data/.htpasswd
# Разрешаем доступ только авторизованным пользователям
Require valid-user
# Закрываем директивы, описывающие секцию
</Location>

# Фиксируем все ошибки виртуального хоста в отдельный
# лог-файл
ErrorLog /var/log/apache2/calendar_error.log
# фиксируем все действия, связанные с доступом
# к WebDAV-каталогу, в отдельный файл, формат файла
# «combined» описан выше (это один из шаблонов в стандартном
# файле конфигурации apache2)
CustomLog /var/log/apache2/calendar-access.log combined
# Закрываем контейнер, описывающий виртуальный хост
</VirtualHost>
```

Конфигурационный файл дополнен, перед запуском веб-сервера проверим его синтаксис:

```
# apache2ctl -t
```

```
Syntax OK
```

Если ответ команды отличается, внимательно читайте вывод, обычно он достаточно информативен и кроме указания строки с ошибкой подсказывает, как ее решить. Например, удалим из директивы CustomLog указание на формат файла журнала:

```
CustomLog /var/log/apache2/calendar-access.log
```

И запустим проверку:

```
# apache2ctl -t
```

```
Syntax error on line 720 of /etc/apache2/apache2.conf:
CustomLog takes two or three arguments, a file name, a custom
log format string or format name, and an optional "env=" clause
(see docs)
```

Нам справедливо замечают, что на 720-й строке ошибка, директива CustomLog принимает 2 обязательных аргумента и один, третий, необязательный.

Исправим ошибку и запустим веб-сервер:

```
/etc/init.d/apache2 start
```

Это можно сделать и так:

```
apache2 -k start
```

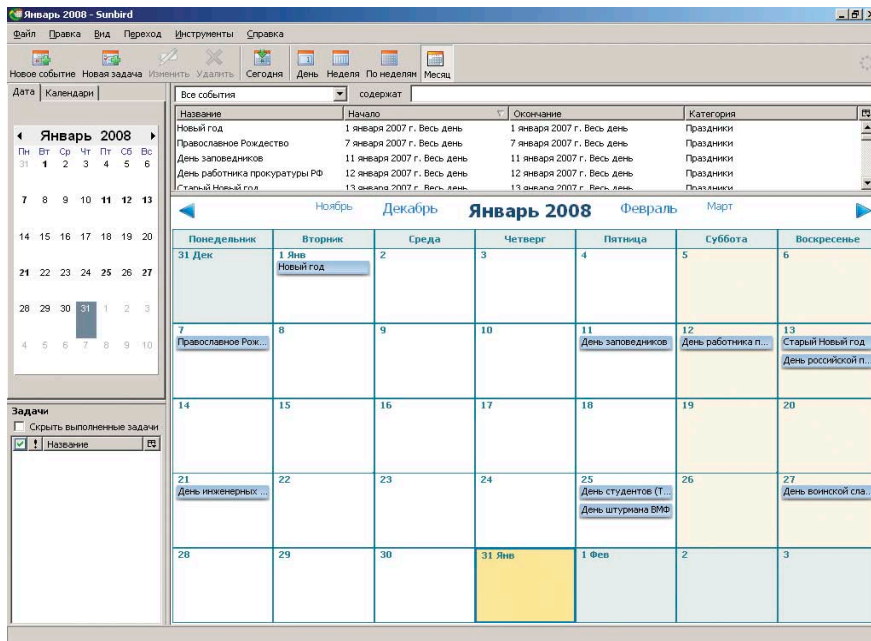



Рисунок 1. Интерфейс календаря Mozilla Sunbird

Если все было сделано правильно, то запускаем клиентов.

Клиенты

Скачиваем с сайта разработчика последнюю версию Sunbird под ОС Windows и устанавливаем ее. Если клиент под управлением ОС семейства Linux, то для установки лучше воспользоваться стандартным менеджером пакетов для работы с репозитарием. После запуска Sunbird подключаем сетевой календарь, пройдя по пунктам: «Файл → Новый календарь → В сети → iCalendar (ICS)», в поле «адрес» введем сетевой путь к календарю вида: <https://server/uploads/RussianHolidays.ics>. В последнем диалоговом окне дадим имя календарю. Если на веб-сервере был использован самоподписной сертификат, при подключении к календарю появятся 2 диалоговых окна, предупреждающих пользователя о том, что невозможно проверить подлинность сертификата. Далее появится окно аутентификации пользователя, то есть пароль будет передан уже зашифрованным. Если все введено верно, можно будет увидеть подобную картину (см. **рис. 1**).

Из настроек клиента, пожалуй, сразу стоит одну изменить: «Инструменты → Настройки → Основные → Обновлять Календарь каждые 30 минут», значение 30 стоит по умолчанию и, на мой взгляд, слишком велико – все-таки календарь должен обновляться чаще. Собственно, на этом все, можно работать, осталось только добавить возможность синхронизации календаря с мобильными устройствами.

Синхронизация с мобильными устройствами

Как уже упоминалось в самом начале, для синхронизации календаря можно воспользоваться двумя программами, на выбор: FinchSync [4] или BirdieSync [5]. Перейдем к рассмотрению и настройке программ:

FinchSync – клиент-серверная программа, написанная на java, предназначена для синхронизации ежеднев-

ника, задач и адресной книги из продукции семейства Mozilla с мобильными устройствами. Первое, что нам понадобится, Java, а точнее Java Runtime Environment (JRE) 5.0, скачать ее можно отсюда [6], файл размером приблизительно 16 Мб. Далее скачаем FinchSync, сервер – jar-файл и клиент, последний существует в двух версиях – для Pocket PC и для SmartPhone, объем всех файлов менее 1 Мб. И последнее подготовительное действие – экспортируем календарь из Sunbird, сохранив его на локальный диск в формате ics. Зачем? Это один из недостатков FinchSync – он не может самостоятельно подключиться к WebDAV-серверу и авторизоваться на нем, но о плюсах и минусах поговорим позже, перейдем к настройке.

Запустим сервер (см. **рис. 2**).

На первой закладке «Activity» будет отображаться ход соединения КПК с ЭВМ, внизу имя и IP-адрес сервера. Перейдем на закладку «Sync Source», нажмем кнопку «Add», в первом диалоговом окне введем имя будущей задачи, далее в «Source» укажем «Mozilla Calendar/Sunbird», нажав «Browse», выберем сохраненный на локальный диск файл календаря, подтвердив выбор, перейдем к закладке «Clients» на главной панели сервера. Нажав «Add», добавим ресурсы, необходимые для передачи клиенту. Запомним поля Name и Password, они обязательны для заполнения и пригодятся нам в дальнейшем. Проверить, работает ли сервер или нет, можно, запустив браузер и набрав адрес http://ip_server:8080/status, где будет предложено ввести логин и пароль сервера для остановки FinchSync.

Перейдем к мобильному клиенту. Перепишем файл FinchSync.cad на КПК, запустим его и перейдем к настройке, пройдя по пунктам «Пуск → Программа → FinchSync → Config → Server → Add» (см. **рис. 3**). Заполним поле Host/IP, порт, Login и Password. Последние два – это Name и Password, указанные в закладке «Clients» на сервере, далее протестируем соединение, нажав «Test» и «Connect». В ответ на экране должны появиться открытые на сервере ресурсы. Ход соединения можно будет наблюдать на ЭВМ в закладке «Activity». Собственно все, нажав кнопку «Sync it!» в главном диалоговом окне FinchSync на КПК, произведем синхронизацию.

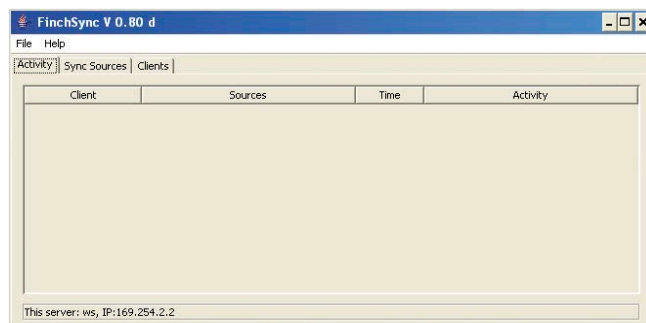


Рисунок 2. Скромный интерфейс сервера FinchSync

К отрицательным моментам данного решения можно отнести сложность настройки для пользователей. Невозможность подключаться к WebDAV-серверу, приходится вручную сохранять файл с календарем из Sunbird. И последнее, явное и очевидное – при описанной конфигурации (с WebDAV-сервером) синхронизация данных может быть выполнена только в одну сторону, с WebDAV-сервера на КПК, зачастую этого достаточно, но все же не всегда. Поэтому переходим к рассмотрению второй программы синхронизации.

Скачав дистрибутив BirdieSync размером 3,5 Мб, запустим его на ЭВМ. Программа после некоторых простых вопросов самостоятельно интегрируется в Microsoft ActiveSync и КПК. Далее запустим Sunbird и ответим также на несколько простых вопросов о необходимых для импорта данных: адресной книги, задач, ежедневника или почты из Mozilla Thunderbird. В настройке и работе программа очень проста и прозрачна для пользователей. В любом случае, что бы ни использовалось – BirdieSync или FinchSync, на КПК и смартфоны будут перенесены необходимые задачи (см. рис. 4).

Но теперь о минусах BirdieSync, главный из них заключается в том, что программа не бесплатна, ее можно использовать для тестирования в течение 21 дня, а потом заплатить 19,95 евро. Программа за время тестирования иногда переставала работать, но помогал простой перезапуск ActiveSync.

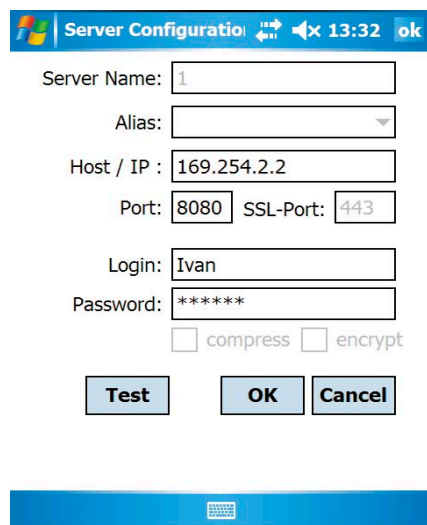


Рисунок 3. Интерфейс настройки клиента FinchSync на КПК

Пара слов о iCalendar

iCalendar – это стандарт rfc2445, опубликованный в ноябре 1998 года в связи с необходимостью создания единого, открытого формата календаря и планировщика задач. Очень многие путают данный формат с достаточно знаменитой программой iCal от Apple inc, но данная программа появилась только через 4 года – 17 июля 2002 года. Авторы стандарта – Frank Dawson из мало известной Lotus Development Corporation (эту компанию в 1995 году купила IBM) и Derik Stenerson из знакомой почти всем Microsoft Corporation. Разработчики при создании новой спецификации основывались на уже ранее используемом

стандарте vCalendar. Для групповой работы изначально предполагалось использовать электронную почту (пересылая сообщения подобно vCard-визиткам), но стандарт разработан, чтобы быть независимым от транспортного протокола, поэтому теперь это делается либо с помощью WebDAV, либо с помощью SyncML. Также есть возможность публиковать единичные события с помощью hCalendar (сокращение от HTML iCalendar). Так как передача информации изначально планировалась через электронную почту, то был обозначен новый MIME-тип – text/calendar. Расширения файлов календарей могут быть не только .ics, но и .ical, .ifb, .icalendar.

В целом больших сложностей с программой не возникало.

Подводя итог рассмотрения этих двух программ, нужно сказать, что в зависимости от поставленных задач может быть использована либо одна, либо другая. При внедрении данного программного комплекса допускалась синхронизация в одну сторону, поэтому ни один из пользователей не выбрал BirdieSync. Сложность настройки FinchSync была уменьшена написанием короткой инструкции, часть которой и вошла в данный материал.

Итоги

Средство групповой работы было сделано, почти полностью реализовав поставленные задачи. Конечно, пройдет время, и обязательно возникнут новые потребности у пользователей, и они по возможности будут удовлет-

ворены. Продукция Mozilla и раньше показывала себя с хорошей стороны, внедрение Sunbird дополнительно подтвердило положительные стороны данного OpenSource-продукта. Также не могу не упомянуть прекрасную документацию по веб-серверу Apache2, расположенную по адресу [7], к сожалению, в основном на английском языке, но прекрасно структурированную и понятную.

Пожалуй, добавлю одно наблюдение. На предприятии, где был внедрен Sunbird, в конференц-зале, где проводятся внутренние совещания компании, на стене висит двухметровая белая доска, раньше она была вся исписана планами на месяц вперед, теперь она пуста, на ней висит прикрепленный на магнит распечатанный лист бумаги формата А3, отображающий деловой месяц компании. Маркеры в стороне. Внедрение удалось.

Удачи на IT-фронтах!

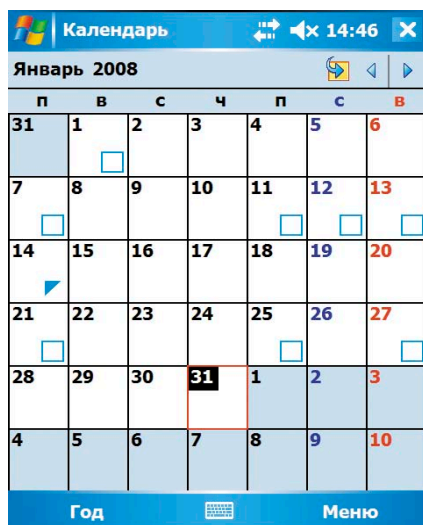


Рисунок 4. Стандартный интерфейс календаря в Microsoft Windows Mobile

1. <http://www.rg.ru/zakon/2008/01/01.html> – Российская газета, «Законы, вступившие в силу 01 января 2008 г.».
2. <http://www.mozilla.ru> – сайт «Mozilla Russia».
3. <http://rfc-editor.org> – RFC-Editor Webpage.
4. <http://www.finchsync.com> – сайт проекта FinchSync.
5. <http://www.birdiesync.com> – сайт проекта BirdieSync.
6. <http://java.sun.com/j2se/1.5.0/download.jsp> – страница загрузки JRE.
7. <http://httpd.apache.org/docs/2.0> – страница, посвященная документации веб-сервера Apache2.