

Строим персональный SyncML-сервис синхронизации

Максим Иргизнов

Контакты, задачи, календарь в мобильном телефоне, КПК или персональном компьютере – возможно ли их объединить в единую информационную среду с удобным доступом? Можно ли получить к ней оперативный доступ, находясь в поездке? Теперь на эти вопросы есть ответ.

Нам очень часто необходимо иметь единый центр хранения пользовательских данных (контакты, календарь, расписания задач, заметки), как на работе, так и дома. Как правило, эти данные необходимы на разных устройствах или рабочих местах. В таких случаях локальная синхронизация с помощью ПО производителя устройства становится очень неудобной.

Технология SyncML призвана помочь в решении этой проблемы. Вы можете создать единый центр синхронизации с удаленным доступом и поль-

зоваться своим органайзером в любом удобном месте, получить удобные средства создания и редактирования различной информации. Также единый центр хранения данных обеспечивает соответствующее резервирование и быстрое восстановление информации.

В Интернете существуют различные сервисы (см. ссылки по теме) для синхронизации информации, но они, как правило, платные либо имеют ограничения, а доверять стороннему серверу приватную информацию часто желания нет.

В качестве рабочей станции я использую самую последнюю версию FreeBSD 7.0-STABLE, но замечу, что все описанное можно также проделать на любой другой UNIX-подобной системе с соответствующими поправками к разделу «Установка».

Этап 1. Перед установкой

Для работы с SyncML потребуется (подробное описание устанавливаемой библиотеки можно получить в файле pkg-descr) установка следующих библиотек и программ из коллекции «портов» FreeBSD:

- /usr/ports/devel/cmake;
- /usr/ports/textproc/wbxml2;
- /usr/ports/devel/libsoup;
- /usr/ports/comms/openobex (не обязательно, если не будете делать локальную синхронизацию по протоколу OBEX).

Устанавливаете их вашим любимым способом: «make install, portinstall, etc».

Для работы понадобятся следующие модули (текущий стабильный релиз):

- <http://opensync.org/download/releases/0.36/libopensync-0.36.tar.bz2>
- <http://opensync.org/download/releases/0.36/libopensync-plugin-file-0.36.tar.bz2>
- <http://opensync.org/download/releases/0.36/libopensync-plugin-irmc-0.36.tar.bz2>
- <http://opensync.org/download/releases/0.36/libopensync-plugin-syncml-0.36.tar.bz2>
- <http://opensync.org/download/releases/0.36/libopensync-plugin-vformat-0.36.tar.bz2>
- <http://libsincml.opensync.org/download/releases/0.4.6/libsincml-0.4.6.tar.bz2>

Инструмент, с помощью которого мы будем производить синхронизацию: <http://opensync.org/download/releases/0.36/msyncntool-0.36.tar.bz2> (существует и GUI-версия этой утилиты – <http://opensync.org/attachment/wiki/download/multisync-gui-0.91.0.tar.gz>). Скачиваем их к себе в домашний каталог любимым download-инструментом.

Этап 2. Установка

Итак, сборка ПО из портов (рекомендую обновить их на текущую дату) закончена, приступим к самому главному.

Для простоты всю работу будем производить в домашней директории пользователя (во всех примерах это будет домашний каталог /home/xeon).

Внимание: для корректной сборки необходимо проверить, чтобы glib2 (/usr/ports/devel/glib20, при отсутствии устанавливается как зависимость libsoup) был собран с поддержкой posix thread, например, так:

```
# ldd /usr/local/lib/libgthread-2.0.so
libthr.so.3 => /lib/libthr.so.3
```

Выделенная красным строка (как вариант там может быть libpthread.so.2) показывает наличие поддержки потоков (см. рис. 1).

Внимание: наличие поддержки потоков является обязательным условием для продолжения сборки библиотек.

Первым делом распаковываем и устанавливаем пакет libsincml:

```
# tar xzf libsincml-0.4.6.bz2
# cd libsincml-0.4.6
# mkdir build
# cd build
# cmake -DCMAKE_INSTALL_PREFIX=/usr/local -DENABLE_OBEX=no ../
```

Используя свой любимый редактор, поправьте файл libsincml-1.0.pc, добавив в конец строки «Cflags: -I\$(includedir)» следующее:

```
-I/usr/local/include
```

Сохраняем и устанавливаем библиотеку:

```
# make install
```

Далее – libopensync – основная библиотека, реализующая синхронизацию:

```
# tar xzf libopensync-0.36.tar.bz2
# cd libopensync-0.36
# mkdir build
# cd build
# cmake -DCMAKE_INSTALL_PREFIX=/usr/local ../
# make
# make install
```

Коротко об устанавливаемых плагинах:

- **plugin-vformat** – содержит основные функции по работе с различными форматами синхронизируемой информации.
- **plugin-syncml** – предоставляет реализацию доступа к протоколу SyncML посредством простейших интерфейсов, таких как http, obex. Я буду использовать в работе простейший http-сервер, идущий в составе плагина и не требующий установки стороннего ПО.
- **plugin-file** – обеспечивает синхронизацию данных, расположенных на файловой системе.

Приступим к сборке минимально необходимых плагинов, начнем с libopensync-plugin-file:

```
# tar xzf libopensync-plugin-file-0.36.tar.bz2
# cd libopensync-plugin-file-0.36
# mkdir build
# cd build
# cmake -DCMAKE_INSTALL_PREFIX=/usr/local ../
# make
# make install
```

Повторяем те же действия для плагинов libopensync-plugin-syncml и libopensync-plugin-vformat. При сборке vformat есть одна особенность: необходимо скопировать файл /usr/local/include/iconv.h в каталог libopensync-plugin-vformat-0.36/src и поправить файл libopensync-plugin-vformat-0.36/src/vformat.c, в нем мы заменяем «#include <iconv.h>» на «iconv.h» и устанавливаем плагин.

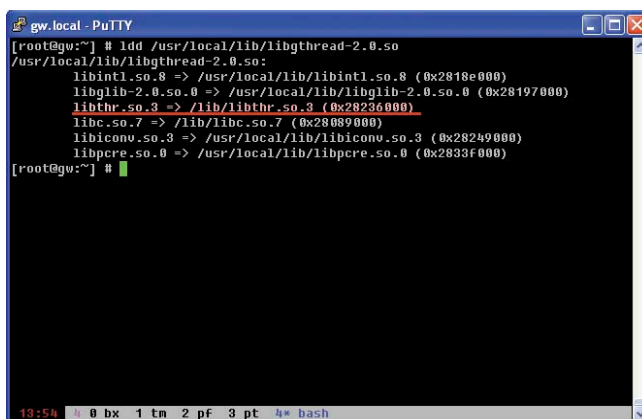


Рисунок 1. Результат команды ldd, показывающий наличие (выделенная строка) поддержки потоков в glib

```

gw.local - PuTTY
[xeon@gw:~] $ msyncntool --listplugins
Available plugins:
syncml-http-server
syncml-http-client
syncml-obex-client
file-sync
file-sync
[xeon@gw:~] $
  
```

Рисунок 2. Просмотр установленных плагинов

Аналогично собираем наш основной инструмент синхронизации msyncntool, после сборки которого установку можно считать законченной и приступить к настройке.

Этап 3. Настройка

Заходим под пользователем, под каким будет работать наша синхронизация. Проверяем, все ли у нас установилось корректно, результат должен быть как минимум такой:

```
$ msyncntool --listplugins
```

```

Available plugins:
syncml-http-server
syncml-http-client
file-sync
  
```

Результат работы команды представлен на **рис. 2**.

По команде «msyncntool --listformats» мы получим список форматов, там должны присутствовать: vcard21(30), vevent20(10), vnote11, vtodo20.

Все в порядке, можно продолжать.

Создаем группу для синхронизации и подключаем к ней наши плагины:

```

$ msyncntool --addgroup mysync
$ msyncntool --addmember mysync syncml-http-server
$ msyncntool --addmember mysync file-sync
  
```

Проконтролировать результат работы данных команд можно так:

```
$ msyncntool --showgroup mysync
```

```

Groupname: mysync
Member 1: syncml-http-server
No Configuration found: Member has not been configured
Member 2: file-sync
No Configuration found: Member has not been configured
  
```

Результат работы команды говорит нам о том, что мы не сконфигурировали членов созданной группы. Сконфигурировать каждого члена созданной нами группы очень просто с помощью такой команды:

```
$ msyncntool --configure mysync 1
```

Откроется ваш редактор по умолчанию с конфигурацией (с комментариями по каждому из параметров) в формате XML.

Для члена syncml-http-server нам достаточно указать URL, на котором будет работать сервер, и port в соответствующих местах конфигурационного файла

```

<url>10.20.30.40</url>
<port>1234</port>
  
```

Замечание: не забудем открыть указанный в конфигурации порт (протокол TCP) на своем брандмауэре. Допустимо использовать 80-й порт в конфигурации, если это не вызывает конфликта с установленным веб-сервером.

Также в конфиге следует проверить (имена баз чувствительны к регистру символов) описания баз данных, к которым будет подключаться клиент. Для телефонов Sony Ericsson часть конфигурации выглядит так:

```

<!-- Contact database -->
<database>
  <name>Contacts</name>
  <objtype>contact</objtype>
  <objformat>vcard21</objformat>
</database>

<!-- Calendar database -->
<database>
  <name>Calendar</name>
  <objtype>event</objtype>
  <objformat>vevent20</objformat>
</database>

<!-- Note database -->
<database>
  <name>Notes</name>
  <objtype>note</objtype>
  <objformat>vnote11</objformat>
</database>

<!-- Tasks database -->
<database>
  <name>Tasks</name>
  <objtype>todo</objtype>
  <objformat>vtodo20</objformat>
</database>
  
```

Замечание: выяснить необходимые версии форматов можно, отправив через Bluetooth-соединение, например, контакт, и в полученном файле посмотреть на строку VERSION. При работе с базой допускается версия выше, чем есть в синхронизируемом устройстве.

Внимание: после запуска сервиса и его отладки вы должны установить username и password в члене syncml-http-server созданной группы в целях повышения безопасности!

Таким же образом конфигурируем второго мембера, в его конфигурации в данном блоке прописываем реальный путь:

```

<directory>
  <path>/home/xeon/.opensync</path>
  <objtype>data</objtype>
</directory>
  
```

Проверяем результат нашей конфигурации:

```
$ msyncntool --discover mysync
```

```

Discovered Objtypes:
contact
  Format: vcard21
  conversion config: (null)
event
  Format: vevent20
  conversion config: (null)
  
```

```

note
  Format: vnote11
  conversion config: (null)
Discovered Objtypes:
  contact
  Format: vcard21
  conversion config: (null)

```

Вот, собственно, и все.

Этап 4. Синхронизация

Вот мы и готовы запустить команду «msyncntool --sync mysync» и подключиться клиентом к нашему серверу.

Отдельные устройства могут потребовать иной формат команды:

```
msyncntool --sync mysync --slow-sync contact
```

где contact – тип (посмотреть типы можно через команду --discover) синхронизируемого объекта (см. рис. 3).

Возьмем для синхронизации современную модель телефона Sony Ericsson (на других телефонах настройки аналогичны). В нем необходимо произвести следующие настройки: «Меню → Параметры → Связь → Синхронизация». Там выбрать:

- **Новая учетная запись** – тип SyncML;
- **Имя** – произвольное.

На вкладке «Общие» указать:

- **Адрес сервера** – http://10.20.30.40:1234/.
- **Имя\пароль** – пока не заполняем.

На вкладке «Приложения» ставим галочки напротив «Контакты» и «Календарь».

На вкладке «Параметры приложения» выбираем «Контакты», вписываем:

- **Имя баз данных** – contacts;
- **Логин\пароль** – пустые.

Также для «Календаря»:

- **Имя баз данных** – calendar;
- **Задания** – tasks;
- **Заметки** – notes.

Удобные имена для названия баз (к сожалению, только в латинице) вы можете сконфигурировать в члене syncml-

```

gw.local - PuTTY
[xeon@gw:~]$ msyncntool --sync contacts --slow-sync contact
Synchronizing group "contacts" [slow sync]
The previous synchronization was unclear. Slow-syncing
contact sink of member 2 of type file-sync just connected
Main sink of member 2 of type file-sync just connected

```

Рисунок 3. Успешный запуск и ожидание подключения клиента

http-server, сохраняем учетную запись и пробуем. Если все заработало успешно, поздравляю!

На рабочей станции под управлением *nix синхронизацию можно проводить, например, с почтовым клиентом Evolution2. Для этого скачиваем и устанавливаем по приведенной в разделе «Установка» схеме плагин libopensync-plugin-evolution2:

```

$ msyncntool --addmember mysync evo2-sync
$ msyncntool --configure mysync 3

```

В файле конфигурации прописываем пути:

```

<config>
  <address_path>file:///home/xeon/.evolution/addressbook/ ↵
    local/system</address_path>
  <calendar_path>file:///home/xeon/.evolution/calendar/ ↵
    local/system</calendar_path>
  <tasks_path>file:///home/xeon/.evolution/tasks/local/ ↵
    system</tasks_path>
</config>

```

Таким же образом устанавливаются и настраиваются плагины для синхронизации с Motorola phones, KDE PIM, Palm, Google Calendar, а также по протоколу irmc.

Популярные телефоны с этой технологией:

- Nokia 3250, 6233, N70;
- Sony Ericsson K790i, K750i, Z750i.

Хочется добавить, что объединение в группу различных типов и методов синхронизации позволяет нам с лёгкостью синхронизировать различные устройства между собой, выполняя единственную команду (достаточно подключить в группу новый (дополнительный) плагин и сконфигурировать описанным выше способом).

Замечания

Для выполнения операции резервного копирования достаточно сделать копию каталога /home/xeon/opensync/ любым удобным способом, например:

```
tar -jcf opensync.tbz .opensync/
```

В рассмотренном способе синхронизация выполняется посредством протокола http, что позволяет очень легко перехватить и подсмотреть ваши данные. Решить проблему можно, перейдя на защищенные https-соединения, которые будут осуществляться, например, с ssl-прокси-сервером.

Используемые в статье программы и примеры можно взять за основу для построения собственного публичного сервиса синхронизации контактной информации.

Удачи! 🍀

1. <http://www.syncml.org> – спецификации по SyncML.
2. <http://www.opensync.org/wiki/SetupGuide> – руководство по установке.
3. <http://ru.wikipedia.org/wiki/SyncML> – информация об SyncML.
4. <http://www.funambol.com> – инструменты синхронизации для различных платформ.
5. <http://nazarov.com/wp/2007/10/24/107> – большой список сервисов для синхронизации SyncML.
6. <http://www.egroupware.org> – сервер групповой работы с поддержкой SyncML.