Linux & Palm in sync
CROSS TALK

MICHAEL TEPPERIS-VON DER OHE

One important design criterion of the Palm PDA is that it regularly has recourse to co-operation with a powerful computer. A regular synchronisation of the data from the mobile organiser with your home Linux PC ensures that no essential dates or data fall by the wayside. This article presents a few of these Palm synchronisation tools for Linux.

The most commonly used software package for connection between the PalmPilot and Linux is Pilot-link. This consists of a whole bundle of individual programs and provides a corresponding Linux tool for every important task for the PalmPilot owner. Programs can be started from the command line with the specification of additional parameters. But there are also a few graphical X11 applications, enabling data synchronisation with the PalmPilot by utilising the Pilot-link console tools. More about this later in this article. For a while it went very quiet on the development front for the Pilot-link packet. But there is now some active work going on in terms of expansions and improvements. At present, work is proceeding on version 0.95 of the current CVS-server of Pilot-link (*http://cvs.gnu-designs.com/cgi-bin/vcvs/viewcvs.cgi/pilot-link/?cvsroot=Open+Palm+Group*).

Pilot-link

I'm now going to describe the installation of the Pilot-link source tarball for Red Hat. But if you are a Linux newbie, in case of doubt you would do better to install a binary package created for your distribution, such as for example in Debian´s *deb*, or in one of the distribution-specific *rpm* formats. In fact, the current distributions usually include only the previous version 0.9.3, but this can be used without worry. Install these binary packages in the usual way:

```
apt-get install pilot-link
```

or with

```
rpm -i pilot-link.0.9.3.rpm
```

Please obtain the exact file name first from your distribution.

Installation from source

First, you should unpack the tar archive with

```
tar xvfz pilot-link.0.9.5.tar.gz
```

After switching to the directory thus created, *pilot-link.0.9.5*, the usual three steps follow:

```
./configure
make
su (+root password)
make install
```

If you wish to install the software somewhere other than */usr/local/* then you must specify this to the *configure* script as an argument, for example:

```
configure –prefix=/usr/pilot
```

Before you install the package with *make install* you should test the converted programs locally. *pilot-xfer* (see below) is probably the best for this.

Conduits

Most programs work like conduits – they transfer data between the Palm and the workstation computer. Palm Inc. has defined various types of conduits. The simplest type just copies data in either direction between Palm and workstation. The more intelligent variant compares individual datasets and merely copies the datasets, which are new or amended for the other side between Palm and workstation. Both variants can be found in the Pilot-link packet. After starting, the programs usually come up with a request to place the Palm in the cradle and to press the Hotsync button. The most important individual programs should now be displayed.

Physical connection

Before exchanging data, you must decide how communication between Palm and Linux computer is to be effected. This contact can be made in one of three ways: serially, via USB or via IrDA. In most cases data exchange with a Linux computer is made via a serial port. To do this, either the cable of the cradle or a corresponding replacement cable is used. USB is becoming increasingly common. There are now cradles for the Palm with a USB plug instead of the 9-pin serial plug, or again, serial-to-USB-adapters (The Visor from Handspring now comes with USB). Unlike the previously mentioned types of connection, IrDA works without a cable. There are two software packages under Linux for this, with different ranges of tasks. While *openobex* is concerned only with the transfer of objects, meaning files in this case, the Linux-IrDA stack generally allows communication via IrDA. But now to present the individual command line tools in the Pilot-link packet.

Pilot-xfer

Pilot-xfer can be used to download and install *.prc* and *.pdb* files onto the PalmPilot. It is also possible to list all files installed on the PalmPilot. There are also sync, backup and restore functions, to back up and/or play back all the files on a PalmPilot. As the standard transfer rate, Pilot-xfer selects 9600 baud. By setting the environmental variable *PILOTRATE* this can be changed, e.g.: *PILOTRATE=57600* (The maximum 57,600 baud). The port used can be defined with the environmental variable *PILOTPORT*: *PILOTPORT=/dev/ttyS1* (Standard COM 2). There is in fact the command option for setting the device to be used, but a corresponding option for the transfer rate is apparently not yet available.

Install-memo

With install-memo text files can be installed as memos. The command

```
install-memo /dev/ttyS1 -c Personal MyText.txt
```

creates a new memo in the category *Personal* with the content of the file *MyText.txt*

Install-user

With install-user you can define and read out the user name and the user-ID of your PalmPilot.

```
install-user /dev/ttyS1
```

reads out the user name and the user-ID.

```
install-user /dev/ttyS1 "Colin R. Miller"
```

defines new user name (Colin R. Miller).

```
install-user /dev/ttyS1 "Colin" 4711
```

defines new user name (Colin) and user ID (4711).

Memos

The following command fetches all memos from the Palm and shows them in standard mailbox format:

```
memos /dev/pilot
```

Pilot-addresses

This command is responsible for the transfer of address data to and from the Palm. The command

```
pilot-addresses /dev/ttyS0 -w storage.file
```

fetches the address data from the Palm and writes them into a Linux file.

Table: Principle Parameters of the Command line tools Pilot-xfer

Command	Function
pilot-xfer -p /dev/ttyS0 -i Conqueror.prc	installation of a program/ a database file (PalmPilot connects to COM 1)
pilot-xfer -p /dev/ttyC4 -f MemoDB	download a database file (PalmPilot connects to Port 4 of a Cyclades multiport card)
pilot-xfer -l	listing of all database files, the port taken from the environmental variable (see above) or */dev/pilot* is used
pilot-xfer -b my_backup	backup of PalmPilots in the directory *my_backup*, and if the directory does not exist, it is created
pilot-xfer -r my_backup	restore a backup from the directory *my_backup*
pilot-xfer -v	outputs the version number
pilot-xfer -h	lists available command options

Conversely, with

```
pilot-addresses /dev/ttyS0 -r storage.file
```

you can write address data from a Linux file to the Palm.

Proprietary formats

Most (commercial) desktop applications, regardless of which operating system they are running on, use a proprietary format to store data. This is no different on the PalmPilot on regular workstations. For example, if instead of the standard mail program, you use one of the competing products on the Palm, you usually no longer have the option of looking at or even editing the data managed by the program with the original program. The same thing applies if you have transferred the data onto your home PC. If you use the original Palm desktop for diary data or memos, this data cannot be edited with comparable desktop applications. This is a highly unsatisfactory situation. Sometimes you want, or indeed must, synchronise diary details with other diary details (work groups, etc.). Equally, you will of course wish to continue to use address data in text editing or even pass them on to other systems – to a mobile phone for instance. These are the reasons why I don't normally use desktop applications, but am searching for solutions to allow me multiple use of data. My e-mail is on my Linux server, where an IMAP daemon does its duty. I can call up messages and read them at any time with a suitable e-mail client. The address data is in a database on my SQL server, to which the Palm also has access.

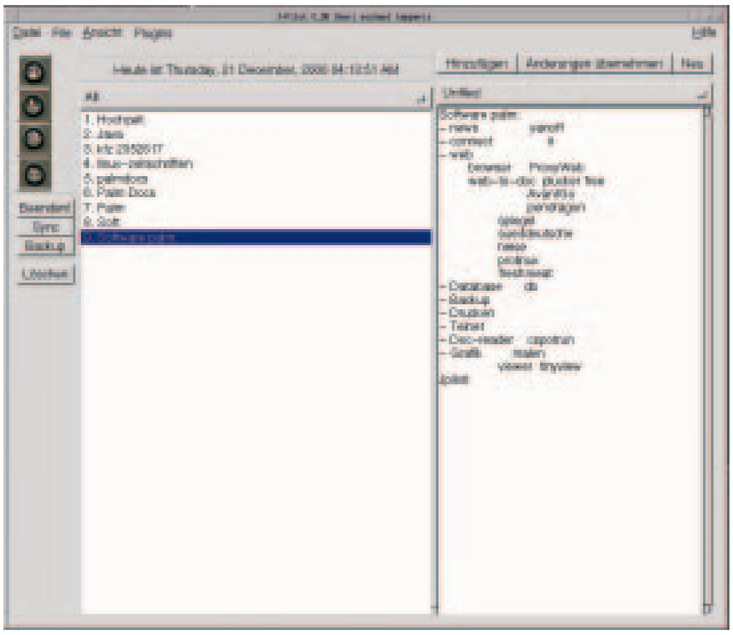

Figure 1: jPilot at work

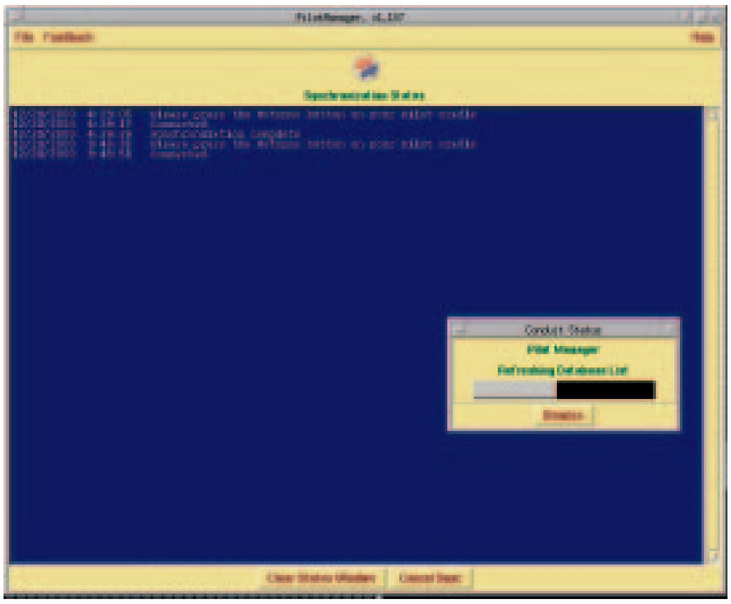

Figure 2: Pilot-Manager during synchronisation

Graphical front ends

Nevertheless, there is a range of desktop applications for Linux that attempt to offer the user what the Palm desktop applications offer. These programs usually offer processing of the following data: addresses, diary, memos, to-do lists and e-mail. I will now present to you the best known graphical front ends for Pilot synchronisation under Linux.

J-Pilot

J-Pilot (*http://jpilot.linuxave.net/*) is a desktop application which relies heavily on the *Palm Desktop* from Palm, Inc. Anyone coming from this Windows/Mac tool will certainly soon feel at home under Linux with J-Pilot. J-Pilot offers the following features: diary, addresses, to-do list, memos, search, install programs, print, synch, backup and plug-ins. The good thing about J-Pilot is the standard range of functions can be expended using plug-ins. At present the following tasks are performed by plug-ins: managing mails, converting Web pages, time-synchronisation and accounts management.

Useful plug-ins

Time synchronisation is done using *sntp* with a timeserver. Converting Web pages is one especially interesting option for making data mobile. There are commercial providers such as AvantGo or Pendragon offering MAL services. This makes it possible to subscribe to the desired Web sites with news. If you have an account for such a service, during the HotSync a connection is made to the remote server and the individual data are combined into a file and copied onto the Palm. The content can be read with an appropriate Palm program. For Linux users, this service is available through the MAL plug-in. Finally, just a couple of details to round off. J-Pilot is based on the Pilot-Link suite, so as to create connections to the PalmPilot. The program was not, as the name seems to hint, written in Java and nor is it a pure

GNOME application, as the use of the GTK+ widget set might suggest.

Gnome-pilot

Gnome-pilot goes well beyond the requirements of a desktop application for the Palm (*http://www.gnome.org/gnome-pilot*). It offers professional management options, such as the management of several PalmPilots. Gnome-Pilot consists of a daemon (*gpilotd*), which monitors one or more interfaces for Pilot connections. Also, thanks to its plug-in capability, gnome-pilot offers numerous expansion options. It handles the management of memos, the installation of programs and backup and the synchronisation of e-mails, but also (like J-Pilot) synchronisation with a MAL-server and synchronisation with the output management of the Palm (costs). The components *GnomeCalendar* and *GnomeCard* are also supported from the Gnome-pim package, which offer the management of diary and address data (see also the article on the overview of scheduler software in this issue). There is also a panel applet. This shows the status of synchronisation with the aid of colours. Finally, programs and documents can be installed via Drag & Drop.

PilotManager

PilotManager is presumably the oldest desktop application for the management of Palm data on Unix computers (*http://www.moshpit.org/pilotmgr/*). Fortunately it has been kept up to date. PilotManager regards itself as a Hotsync daemon, designed for expansions. The program was written in Perl and uses Tk extensions. Conduits are available for synchronisation with various diaries, memos, to-do lists, e-mail, address book, and also with a timeserver or a MAL server. There are also solutions for backup and installation. For example there are the unique options for synchronising address data with the BBDB address book from Emacs and support for XML.

KPilot

KPilot (*http://www.slac.com/pilone/kpilot_home/*) is a KDE program. The latest version was released in April 1999. This would seem to suggest that Kpilot is no longer maintained. Nevertheless there are some users who are very satisfied with it. KPilot is comparable to J-Pilot. It can manage memos, addresses, file installations and e-mails. And Kpilot also has a plug-in port for expansions.

ColdSync

With ColdSync (*http://www.ooblick.com/software/coldsync/*) also, PalmOS devices can be synchronised with Unix workstations. ColdSync is not a graphical application, though. It is worth mentioning that you can synchronise with ColdSync without even having to be logged on. The whole thing is called up with

```
coldsync -p /dev/ttyS0
```

ColdSync simply synchronises the Palm by default with a backup directory. If possible it performs a so-called ‘Fast Sync’ – i.e. it copies only the data from the workstation onto the Palm, which has changed on the workstation. Obviously, the same applies for the opposite direction. To install a new application on the Palm, the file merely has to be copied to *~/.palm/install/*.
ColdSync conduits offer the following features: Conversion of the Palm address book database into the format of *kab*, to the KDE address book and back. The conduit *to-do text* converts back and forth between the PDB format and simple text format. The procedure is the same with memos and a Kjots file. To-do lists and mails can also be synchronised.

StarOffice

Those who love all-in-one packets and are also especially happy with programs of commercial origin will perhaps find the right desktop application for them in StarOffice (*http://www.sun.com/staroffice*), because this, too, offers the options of synchronising data with the Palm. The following Palm data communicates with the corresponding StarOffice data: *To-do* (Palm) with *Tasks* (StarOffice Schedule, see also article on Scheduler software in this issue), *Diary* (Palm) with *Events* (StarOffice Schedule) and *Addresses* (Palm) with *Address Book* (StarOffice).

jSyncManager

The jSyncManager, written in Java, comes from the OS/2 world (*http://warp.idirect.com/*). According to its own description, though, this is also suitable for any other Java environment. Unfortunately I only discovered this program shortly before going to press for this issue and was therefore unable to test it in time. But if it keeps the promise it makes on the homepage, this is my hot favourite!

Other alternatives

Of course, in addition to the programs presented here, there are some other interesting desktop applications for Palm synchronisation, but to discuss these in detail would take up far too much space for this article. These include JetSync (*http://mega.ist.utl.pt/~frias/jetsync/*), Pyrite (*http://pyrite.linuxave.net/pyrite/*), formerly PalmPython, and the Linux Palm Desktop from Applixware (*http://freshmeat.net/projects/linuxpalmdesktop/homepage/*), and there will certainly be more to come. ■