

Serial Links for Computers

Instant Network

Networking is one of Linux's stronger points. But what happens if you do not have a network? You can always resort to a serial link.

BY ANDREA MÜLLER

You have probably faced this problem before: you need to transfer data between Linux and a Windows computer. If you do not have a network, you are normally forced to use external media. But a null modem cable will provide a simpler and more flexible way to link up the two computers. Of course, you have to make do with a slow data transfer rate of about 11 kbit/s, but apart from that you gain all the benefits a "real" network has to offer. Additionally, this solution is cheaper and easier to set up than a network.

Quick Brew

You will need the right kind of lead for a start. Your local computer shop or an electronics warehouse is probably the best place to look. A "null modem connection set" will probably set you back somewhere in the region of £10. Now attach the lead to the serial ports on both computers, just like you would attach a modem.

You can now start configuring the connection. For Linux you need to setup **pppd** to accept connections via the null modem lead. Assuming that you are using *ttyS0* (the COM1 port), ensure that you are *root* and create a configuration file called */etc/ppp/options.ttyS0*. Listing 1 shows a sample file.

The entries preceded by "#" are remarks designed to explain the options used here. The third line from the bottom is important, as the IP addresses for the server (192.168.0.1) and client (192.168.1.1) are specified there.



Additional details on the individual options are available on the *pppd* manpage. That would normally complete the configuration steps for the Linux end, but unfortunately Windows has a few special requirements. The last line tells *pppd* to use the */etc/ppp/win.chat* script to call **chat** before opening a connection. This is required because Windows insists on introducing itself to its dialog partner before opening a *PC Link*. Windows sends "CLIENT" and expects "CLIENTSERVER" as an answer. To appease Windows you will need to create a script called */etc/ppp/win.chat* with the following contents:

```
TIMEOUT 3600
CLIENT CLIENTSERVER\c
```

The *TIMEOUT* value is how long *chat* should wait for the "CLIENT" string.

Now its time to configure the Windows client. As already mentioned, we will be using the *Direct Cable*

Connection program, which is available in *Start/Programs/Accessories/Direct Cable Connection*. If the *Direct Cable Connection* program is not yet installed, you can install the program by going to *Start/Properties/ControlPanel/Add/Remove Programs/* and selecting *Windows Setup, Connections/Details/Direct Cable Connection*. (For XP select *Start/Programs/Accessories/Communication/New Connection Wizard*, then *Set up an advanced connection* and finally *Connect directly to another computer*.) The first time you start the program, a

GLOSSARY

pppd: The daemon responsible for setting up Point to Point Protocol connections, that is a special kind of TCP/IP connection between two computers. The *pppd* is usually launched whenever you use a modem device to access the Internet.

chat: a program normally used to script modem dialogs. It waits for a string and sends a corresponding answer to the modem.

Listing 1: Configuration file options.ttySo

```
# Dial up configuration for clients via null modem cable

lock          # create lockfile (/var/lock/LCK..ttySO)
noauth        # no authentication
asynctmap 0
crtscts       # enable hardware flow control
local         # this is not a modem
silent        # wait for connections
192.168.0.1:192.168.1.1      # IP addresses server:client
115200        # Line speed -- may need to be reduced for long cables
connect 'chat -v -f /etc/ppp/win.chat'
```

Wizard will help you with the configuration steps. Select *Guest Computer* in the first window and then click on *Next* to confirm. In the next window you will need to enter the serial port the null modem cable is attached to.

The final window prompts you to configure the host computer. The computer will attempt to open a connection immediately after you click *Finish*, so you might like to launch *pppd* on your Linux machine first with:

```
pppd /dev/ttySO nodetach
```

(If you are not using the first com port, you will need to customize the device file to match.)

The “nodetach” flag prevents *pppd* from running as a background daemon and allows you to watch any messages issued. You can now click on *Finish* in the Windows Wizard. A few seconds later, you should see a screen similar to Figure 1.

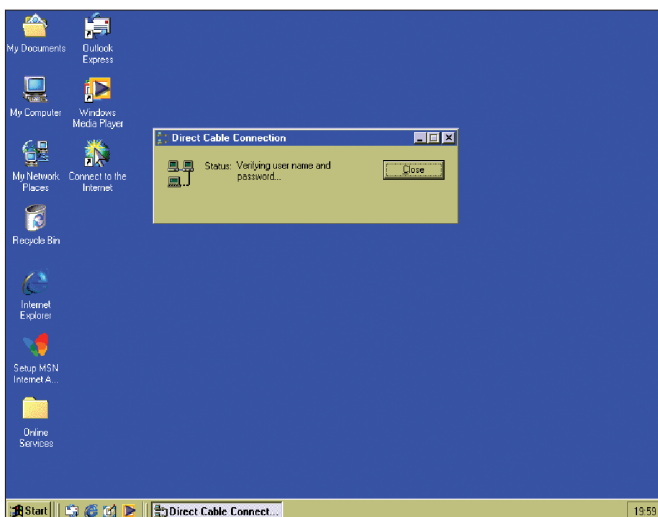


Figure 1: Windows has set up a Direct Cable Connection to the Linux host

Back on Linux you can monitor any activity in the terminal window where you started *pppd*.

```
[andi@gemini andi]$ pppd >
/dev/ttySO nodetach
Serial connection established.
Using interface ppp0
Connect: ppp0 <--> /dev/ttySO
local IP address 192.168.0.1
remote IP address 192.168.1.1
CCP terminated by peer
Compression disabled by peer.
```

To discover whether data transfer is actually occurring, why not

```
ping -c 4 $IP-NUMBER
```

in an *xterm* or in the MS DOS Command Prompt (or simply Command Prompt for the XP inclined), to test the connection? In this example “\$IP-NUMBER” refers to the IP address of the other computer. You do not need the *-c 4*

flag for Windows, as the Windows *ping* only transmits four data packets. The following line in the ping output is interesting:

```
4 packets transmitted, 4 packets
received, 0% packet loss
```

This tells you that the communications link is working, and you can get on with some fine tuning.

Stir Well

Terminate the connection by clicking on *Close* in the Windows in the Direct Cable Connection program window. To avoid having to remember lengthy IP addresses you might like to customize the *hosts* file on both computers. On Linux the file is located in the */etc* directory; Windows supplies a template called *hosts.sam*, which is stored in the Windows directory, and needs to be renamed to *hosts*. After editing, the files could appear as shown in Listing 2.

Incidentally, on Windows XP the *hosts* file should be stored in *c:\windows\system32\drivers\etc*.

Now open the connection again and repeat the connectivity test by typing:

```
ping $COMPUTERNAME
```

using the short form of the computer name. If the output again says *0% packet loss*, you know that name resolution is also working.

To use the services provided by your Linux computer you need to carry out one more configuration task on your Windows computer. If you attempt to access your Linux web server using the Internet Explorer, IE will try to open a dial up connection to your provider. To prevent this happening, you should go to *Start/Properties/Control Panel/Internet Options* and select *Never dial a connection* in the *Connections* tab.

You should also be aware of the Windows alert that appears shortly after opening the connection. This means that Windows has discovered that the server at the other end of the connection does

Listing 2: The Windows *hosts* file

```
#hosts Linux
127.0.0.1      gemini.localdomain  localhost
192.168.1.1   oemcomputer.localdomain oemcomputer

#hosts Windows
127.0.0.1      oemcomputer.localdomain localhost
192.168.0.1   gemini.localdomain  gemini
```

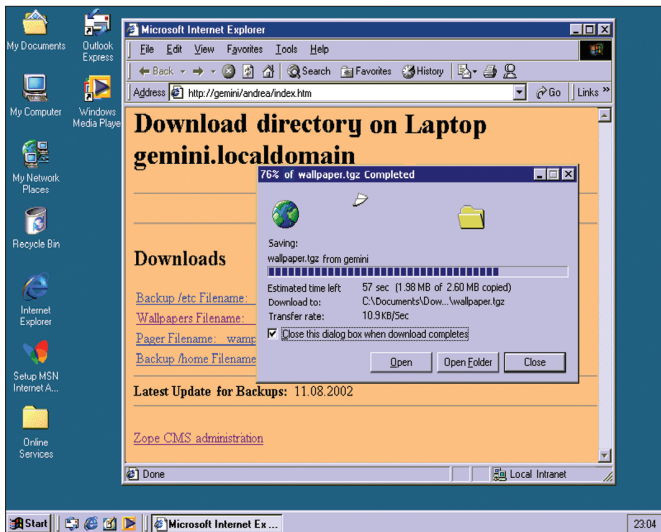


Figure 2: Exchanging HTTP data between Apache and Internet Explorer

not have any shares. You will be prompted to type the name of the host computer to allow Windows to look again. You can click *Cancel* to close the dialog box.

On the Linux side of the connection you may have noticed that error messages started appearing in `/var/log/syslog` after opening the connection. These messages are generated by services that are launched by the `/etc/ppp/ip-up` script, and normally perform useful tasks, such as transferring Mail or collecting News. To maintain the reliability of these services – your MTA

file. But make sure that you create a backup before you attempt to edit the file, as most distributors advise against editing `ip-up`.

```
cp /etc/ppp/ip-up >
/etc/ppp/ip-up.bak
```

The lockfile created by `pppd` will help us prevent `ip-up` from being run. To do so, place the original content of the `ip-up` file in an “if” block that runs your “internet commands” for connections that do not use the serial port where the null modem cable is attached. The example in Listing

Listing 3: Customized `/etc/ppp/ip-up` file

```
# if a lockfile (LCK..ttyS0) exists for the port
# with the
# null modem cable, terminate the script
if [ -f /var/lock/LCK..ttyS0 ]
then
    exit 0
# otherwise run the original script
else
    ORIGINAL CONTENT OF FILE HERE
fi
```

may give up after one or two unsuccessful attempts – you will need to modify the `ip-up`

3 shows you the kind of entries you will need. Add these lines after your distribution’s comment lines (which start with “#”), but before the script commands. It is also important to remember the `fi` at the end of the script, to close the “if” block. You may need to customize the serial port in the line starting with “if”.

Ready to Dine

Now there is nothing to prevent you exchanging data between Linux and Windows, as you can see in Figure 2.

Whether you intend to use HTTP or FTP, or even want to access `leafnode` or `mysql`, is a matter of personal taste, and your requirements. However, when setting up the required services you should ensure that you allow access to your Windows client, but not to the whole internet. The documentation most servers provide includes a section on security issues, and tips on protecting your privacy.

The Direct Cable Connection can be used in the other direction too, allowing your Linux computer to access Windows services. The POP/SMTP/NNTP server *Hamster* is useful for that task. ■

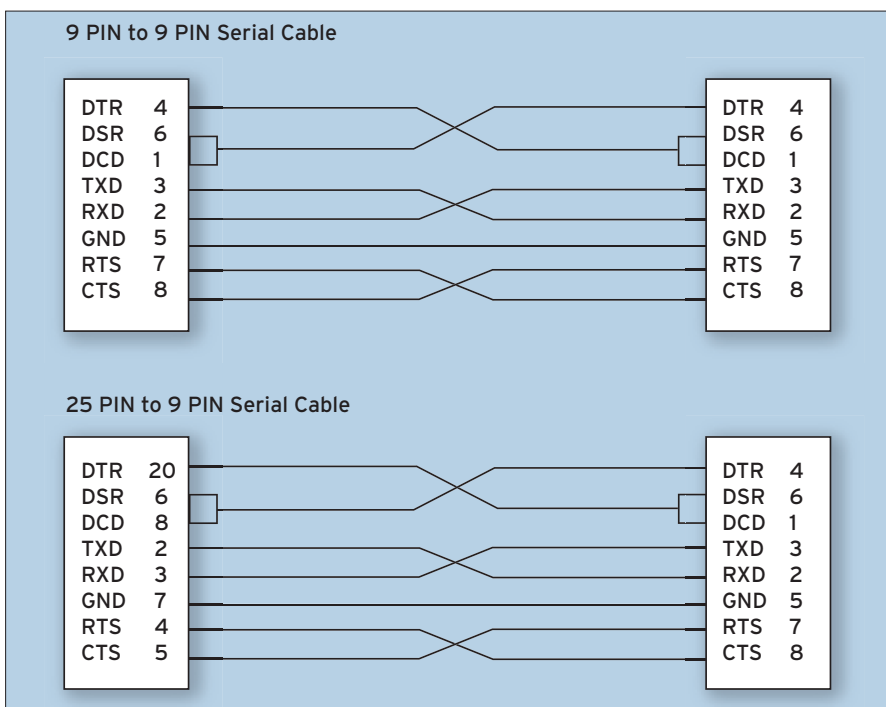


Figure 3: Null modem cable connections.

THE AUTHOR

Andrea Müller is a law student who keeps herself busy with Linux whenever she gets tired of legal theory.



When time permits she additionally likes to take a peek at other operating systems, such as QNX, BeOS and NetBSD, or even tries to polish her Python skills. Apart from Linux and her university career Andrea is interested in literature, European history and cycling.