NCP NETWORK COPY TOOL

CHRIS PERLE

There are thousands of tools and utilities for Linux. "Out of the box" takes the pick of the bunch and each month presents a program which we consider to be indispensable or, perhaps, little known. This time we take a look at the Net copy tool ncp.

To move files from one computer to another over the Net we usually use ***ftp*** or ***scp.*** But what do we do if we wish to do a quick file transfer between two computers or transfer the output of a command line operation to somewhere else? In this case, the utilities mentioned above, with the necessary **servers**, are too cumbersome. Felix von Leitner has probably had this problem as he is the developer of *ncp*.

Unusual zipping

From the *ncp* homepage (*http://www.fefe.de/ncp/*) we obtain the source text of the program, which is available as a ***bzip2*** -zipped ***tar*** archive. As not all *tar* versions offer automatic use of *bzip2*, we will explain how to unzip it as follows:

```
bunzip2 -c ncp-1.0.tar.bz2 | tar xf -
```

Next we must **compile** the program. To do this you only need to run **make**. Then we must install the executable program in the directory */usr/local/bin* and set up two **symbolic links** with *root* rights, as *root* reserves the write rights in system directories such as this.

ftp: *"file transfer protocol", is a widely used protocol for moving files across the Internet. However, as all data (including passwords) is transferred unencrypted and can therefore be intercepted it is most often used to download files from the archives of public FTP servers or upload files to web servers and not to transfer data between two users' systems.*

scp: *"secure copy", is part of the secure shell package ssh, with which files are transferred encrypted.*

Server: *A program offering certain services which client programs can use when they connect to it. Examples of the services offered include http (provided by web servers), ftp and finger.*

bzip2: *is an alternative to gzip, the standard compression program on Linux systems. Often bzip2 compresses data better than gzip.*

tar: *"tape archiver", is the standard archiving program under Unix. It reduces whole directory structure to a single file, which can be written or compressed on magnetic tape (hence the name).*

Compile: A program cannot be executed by the operating system while it is in the form of source code. Only after it has been compiled (converted) into executable code using a compiler is it in a form that can be run by the processor. The advantage of distributing programs in source code form is that they can be compiled to run on different platforms (Intel, Sparc, Alpha ...) as well as making it easy for you to make your own modifications.

```
cd ncp
make
su   (enter root-password)
cp ncp /usr/local/bin
cd /usr/local/bin
ln -s ncp npoll
ln -s ncp npush
exit
```

Different operating modes

ncp can be used in different modes. If it is called up using the name *ncp*, it sends or receives one or several files. As an example, we could send the whole **/etc** directory from *computerA* to *computerB*. To do this, *ncp* is started on *computerB* in server mode and then on *computerA* in client mode:

```
[ComputerB]$ ncp
ncp: server mode.  waiting for connection.

[ComputerA]$ ncp ComputerA /etc
tar: Remove leading `/' from absolute file name in archive.
drwxr-xr-x root/root        0 2000-05-30 09:26 etc/
-rwxr--r--root/root     2096 1999-03-11 18:03 etc/hosts
...
```

If you call up *ncp* under the name *npush* or *npoll* (these are the symbolic links we set up during the installation procedure) it sends or receives the **standard input**.

It is up to us what we feed to the *npush* standard input. As an example, we transfer the content of the *text* directory as a *bzip2*-zipped archive, which is only to be saved and not unzipped on the opposite side:

```
[ComputerA]$ tar cf - text | bzip2 | npush
npush: IPv4 multicast failed, trying IPv4 broadcast

[ComputerB]$ npoll ComputerA > text.tar.bz2
connecting to ::ffff:192.168.0.1
```

Let's break down the command string on *computerA*. First of all *tar* creates an archive using *c* ("create") which goes via "*f -*" to the **standard output**. With the help of the pipe sign **|** this is diverted to *bzip2*, which in turn forwards its output to *npush*. On the opposite side (*computerB*) *npoll* receives the data and outputs it to the standard output, which is redirected to a file by the name of *text.tar.bz2*.

ncp on disk

There is another gem for users with some experience of the command line. *ncp* is very useful in that it can be used to transfer data between two computers with a minimum of requirements. For example, I modified the small Linux distribution *hal91* so that two computers can be connected via a **null modem cable**. All this requires is *hal91* to be booted from the diskette on both computers and the connection to be established using the script *ppp-nullmodem* (which you may have to modify):

```
#!/bin/sh

# Compression for PPP (not absolutely necessary)
insmod /tmp/bsd_comp.o

# establish PPP connection
# Serial interface: /dev/ttyS0 or /dev/ttyS1
# Give as IP address pair:
# Here in the example on computerA: 192.168.0.1:192.168.0.2
# On computerB: 192.168.0.2:192.168.0.1
pppd /dev/ttyS1 115200 asyncmap 0 noauth persist local passive
nodefaultroute 192.168.0.1:192.168.0.2
```

That way, data can be moved to a notebook which has only 8MB of main memory and no network card, even if Linux is not installed. ■

make: *is a program for controlling the sequence of events needed to create an executable program from source code. The configuration file for make (the Makefile) contains information on dependencies between individual program modules, for example.*

Symbolic link: *On Unix file systems users have the option to give a single file several different names. The file blah could also be reached under the name blubb using the command ln -s blah blubb.*

/etc: *The configuration files of many different programs are kept in this directory. It is worth backing up /etc so that all the work you may have done modifying these files is not lost after a disk crash, a careless deletion while logged in as root or in the case of a re-installation. Once again the modular structure of Unix pays off. Instead of having to gather all the files in the /etc directory yourself, ncp leaves this task to tar. This tool is also called up at the receiver's end so that the files can be written there. For security reasons tar refuses to include the* ***absolute path name*** *so that the /etc directory on computerB is not inadvertently overwritten.*

Absolute path name: *A complete indication, starting from the root directory, of the location of a file. For example, the absolute path of a file logs/log.txt in my home directory is /home/chris/logs/log.txt.*

standard input, standard output: *Many command line programs offer the opportunity to omit the name of the input file. If this is done the program reads from the standard input, which is usually the keyboard. If the name of output file is omitted many programs output to the standard output which is normally the terminal.*

|: *The pipe sign (representing a pipe) connects the standard output of one program to the standard input of another. Therefore, several programs can be linked together in a pipeline to form one processing step.*

Null modem cable: *A cable providing a direct connection between two computers via the serial interface. In contrast to normal serial cables, the send and receive lines are connected crosswise.*