

Out of the box IN THE PIPELINE



The program *less*, as an easy-to-use substitute for *more*, will already be familiar to most people. As Christian Perle explains, once equipped with a script, *less* can display more than just mere text files

Depending on which Linux distribution you are using, you will already have noticed that in some configurations the **pager** *less* can show more than just pure text files – **manpages** are automatically formatted or archive contents listed. But what mechanism lies behind this? Is there a normal *less* and a super *less*? No, the features used here are all part of the normal range of functions of *less* and can also be expanded into the bargain.

The **shell script** *lesspipe*, by Wolfgang Friebe, takes advantage of the fact that when *less* starts, it checks the **environment variable** *LESSOPEN*. If this is not empty, its content is interpreted as a program name and *less* leaves it to this tool to open the file to be displayed. Its output is then displayed by *less*. A *LESSOPEN* program such as *lesspipe* thus acts as an input pre-processor.

Piping in

It's not hard to install *lesspipe*. First you need to obtain the archive *lesspipe.tar.gz* from <http://www.desy.de/zeuthen/~friebe/unix/lesspipe.html>. After unpacking the configuration script (which requires a Perl interpreter) it finds and collects a range of auxiliary programs and adapts *lesspipe* to them. If an installed tool is not found, you can help the script on its way by specifying the **full path name** in response to its question "Include code anyway?". The following installation steps are necessary:

Pager A program for displaying a file page by page. Common pagers include *more* and *less*.

Manpage Linux, like all Unix systems, has a kind of online reference manual for the installed programs. This utility is invoked by "man programname", for example "man less".

Shell script A file with shell commands, which are executed automatically. Work steps that recur frequently can be automated well by shell scripts.

Environment variable A variable set in a shell, which can be read by all programs started from this shell.

Out of the box

There are thousands of tools and utilities for Linux. "Out of the box" takes the pick of the bunch and each month suggests a little program, which we feel is either absolutely indispensable or unduly ignored.

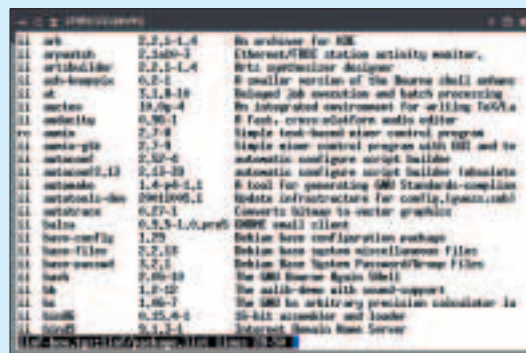


Figure 1

```
tar xzf lesspipe.tar.gz
cd lesspipe-1.32
perl configure
su (enter root password)
make install ; exit
```

After this procedure you need only tell *less* that it should use the script just installed in the */usr/local/bin* directory. To do this enter in the file *.bashrc* in your **home directory** the following line:

```
export LESSOPEN="|/usr/local/bin/lesspipe. 2
sh %s"
```

If, instead of the bash you are using the (t)csh as login shell, you should instead insert the line

```
setenv LESSOPEN "|/usr/local/bin/lesspipe. 2
sh %s"
```

in the file *.cshrc*. On next login the expanded *less* functions will then be at your service.

Show me...

For a simple test you can use any LaTeX short introduction or another *dvi* file which may be installed on your system (and which you can find with *locate l2kurz.dvi*):

```
less l2kurz.dvi
```

By the way, the *lesspipe* script does not recognise file types from their endings, but with the command *file*, which reads a piece of the content of the file and compares it with the entries in its type database.

In Table 1 you will find the file types which *lesspipe* supports, provided the appropriate utility programs are installed. Many of these tools, such as the compressors *gzip* and *bzip2*, are part of the standard installation in most distributions, others (such as *antiword*) often have to be installed later. Non-Debian users will also have to do without the option of viewing Debian archives, because the corresponding tools do not exist for rpm-based distributions.

Nested

Compared to simpler input filters for *less*, *lesspipe* has one especially useful property: it can be used nested. If, for example, you have a *gzip*-compressed tar archive to hand, you can not only see the list of the files contained therein, but also the content of these files – again interpreted by the input filter. In Figure 1 *less* displays the file *linf/package.list* from the archive *linf-box.tgz*. The command for this looked as follows:

```
less linf-box.tgz:linf/package.list
```

Table 1: File types supported

Type	Utility program
.gz, .z	gzip
.bz2	bzip2
.zip	unzip
.tar	tar
Manpage	groff
.a	ar
Dynamic library	nm
Executable program	strings
.rpm, .spm	rpm, rpm2cpio, cpio
.deb	dpkg, dpkg-deb
.doc	antiword
.html	lynx
.pdf	pdftotext
.rtf	unrtf
.dvi	dvi2tty
.ps	ps2ascii, gs

The colon serves here as a separator between the name of the archive file and that of the file contained therein.

The nesting can go even further. Source **rpm** archives contain, in addition to the package information, the original *.tar.gz* archive with the source text of the respective program. If you want to look at the file *gnomo-0.1/README.html* from the archive *gnomo-0.1.tar.gz*, contained in *gnomo-0.1.src.rpm*, enter:

```
less gnomo-0.1.src.rpm:gnomo-0.1.tar.gz:gnomo-0.1/README.html
```

Because the file type is HTML the file is filtered with the text browser *lynx*. If you want to cut out this filter step, when you call up *less* simply place another colon after the file name.

ASCII art

With a bit of know-how about shell programming *lesspipe* can also be expanded – for example to display graphics with the aid of the graphics filter from **netpbm**. The expansion can be found on the cover CD in the form of a **patch**.

In order to apply this patch on the script, install, if necessary, *netpbm*, change again to the directory *lesspipe-1.32* and enter there:

```
patch < path_to_your_cd/ootb/lesspipe-asciiart.diff
perl configure
su (enter root password)
make install ; exit
```

The patch adds four lines to the file *lesspipe.sh.in*, which serves as the model for the configuration script. Since *less* can only display text, *netpbm* filters convert image files into ASCII graphics. In Figures 2 and 3 you can see an example of the graphics capabilities of *less* with this *lesspipe* expansion.

This extends Table 1 by all the graphics formats supported by *netpbm*. Naturally only icons and other simple graphics are recognisable with this method. Additional filter mechanisms are waiting to be built in – maybe a task for rainy summer days?



Figure 2



Figure 3

Full path name The complete directory path to a file. This unequivocally indicates the position of a file in the entire filesystem. The command *top* has for example the full path name */usr/bin/top*.

Home directory This directory is where users will find themselves after logging in. This is also where their personal settings are stored.

rpm With the Red Hat Package Manager (used by Red Hat, Mandrake, SuSE and other distributions) software packages can be cleanly installed and uninstalled. The associated package format is also called rpm.

netpbm A collection of graphics filters, which process various image formats or convert them into different ones. Unlike the Gimp, *netpbm* can cope without a graphical user interface and is therefore ideal for scripts.

Patch A file containing amendments to one or more text files. It is created with the command *diff* and played in with the command *patch*.