Mail Delivery made easy

PROCMAIL

COLIN MURPHY

Last month's article described how to set up and configure a basic mailserver for dial-up machines, giving you control of your email, rather than relying on monolithic mail programs like Netscape Messenger. Fetchmail was charged with the responsibility of collecting email from your ISP, now we have to do something with it to make all of this worthwhile.

Procmail is a small but powerful program, which will not only act as our mail delivery agent – moving the mail from our local mailserver to where a local user can read their messages in the home directory – it will also allow us to filter, or otherwise process, our mail. It will either remove unwanted mail or organise wanted messages so that they are not buried in a pile of spam screaming about get-rich-quick pyramid schemes. As with Postfix and Fetchmail, Procmail is widely available, being supplied with all of the boxed set distributions we've tried.

Procmail is a feature-rich program that no overview could fully describe, all we can do here is give you a taste for the powerful control this application could unleash on your email handling. Here's a list of some of the things it can do as described in the documentation that comes with procmail:

Event driven (invoked automagically when mail arrives)

Does not use *any* temporary files

Uses standard egrep regular expressions

Poses a very low impact on your system's resources (it's 1.4 times faster than the average */bin/mail* in user-CPU time)

Allows for very easy-to-use, yes/no decisions on where the mail should go (can take the size of the mail into consideration)

Also allows for neural net-type weighted scoring of mails

Filters, delivers and forwards mail *reliably*

Provides a reliable hook (you might even say anchor) for any programs or shell scripts you may wish to start upon mail arrival

Supports four mailfolder standards: single file folders (standard and non-standard VNIX format), directory folders that contain one file per message, or the similar MH directory folders (numbered files)

Native support for */var/spool/mail/b/a/bar* type mailspools

Variable assignment and substitution is an extremely complete subset of the standard */bin/sh* syntax

Provides a mail log file, which logs all mail arrival, shows in summary whence it came, what it

was about, where it went (what folder) and how long (in bytes) it was

Uses this log file to display a wide range of diagnostic and error messages (if something goes wrong)

Does not impose *any* limits on line lengths, mail length (as long as memory permits), or the use of any character (any 8-bit character, including '\0' is allowed) in the mail

It has man pages (boy, does it have man pages)

Procmail can be used as a local delivery agent with comsat/biff support (*fully* downwards compatible with */bin/mail*); in which case it can heal your system mailbox, if something messes up the permissions

Secure system mailbox handling (contrary to several well-known */bin/mail* implementations)

Provides for a controlled execution of programs and scripts from the aliases file (i.e. under defined user ids)

Allows you to painlessly shift the system mailboxes into the users' home directories

Procmail calls on .procmailrc, a file in the users' home directory, which you will need to write with your favourite text editor. In its simplest form, you just need to set up some default variables so that it knows where to find and post its work, it needs to look something like this:

```
PATH=$HOME/bin:/usr/bin:/usr/ucb:/bin:/usr/
local/bin:.
MAILDIR=$HOME/Mail        # You'd better
 make sure it exists
DEFAULT=$MAILDIR/mbox
LOGFILE=$MAILDIR/from
LOCKFILE=$HOME/.lockmail
```

You will need to confirm that the directories are actually available otherwise your precious email may go missing. With the latest Procmail versions, it is mandatory that this file is not writable by any user other than yourself, so you may need to run:

```
chmod go-w ~/.procmailrc
```

Procmail compares every email it has against this file. As it stands, all the emails will move your mail into the default standard mail directory, not very exciting so to this file we should add some filtering recipes:

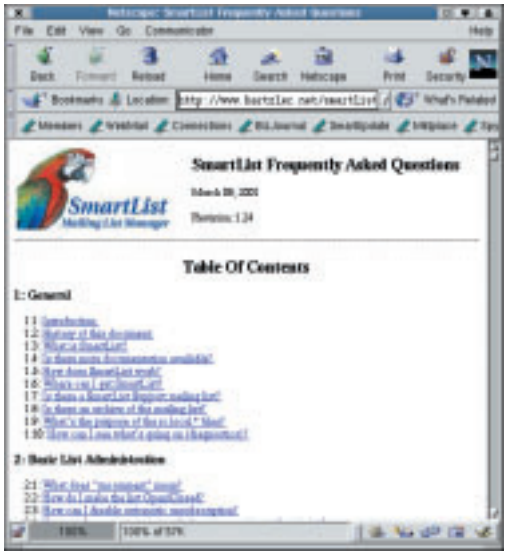
```
:0              # Anything from thf
* ^From.*thf@somewhere.someplace
todd            # will go to $MAILDIR/todd
```

This is very straightforward. The :0 signifies the start of a new recipe, on the second line the * tells us that this is a conditional command, the "From" header in the email is compared to the text expressions, the third line shows the directory to which the email would be sent should the condition about prove true, if we have remembered to create the directory first! So, emails from *thf@somewhere.someplace* and *123thf@somewhere.someplace* should all now appear in the todd subfolder in KMail

Here are some more examples:

```
:0                      # Anything from
 people at uunet
* ^From.*@uunet
uunetbox                # will go to
 $MAILDIR/uunetbox
```

A quick-start tutorial



[left]
faqs galore

[right]
Procmail.org is always worth a look

```
:0                          # Anything from➥
 Henry
* ^From.*henry
henries                     # will go to➥
 $MAILDIR/henries

# Anything that has not been delivered by➥
 now will go to $DEFAULT
# using LOCKFILE=$DEFAULT$LOCKEXT
```

These mail subdirectories will appear in the mail user agent, KMail for instance, as subfolders. Should you be using some other MUA, you should bear in mind that the mail folder used by default might be different. Netscape, for example, uses *$HOME/nsmail* by default for its mail folder, so that's what you would set the MAILDIR variable to.

This still hasn't done anything outstanding though, nothing that we couldn't have achieved using the built-in filtering routines in programs like Netscape.

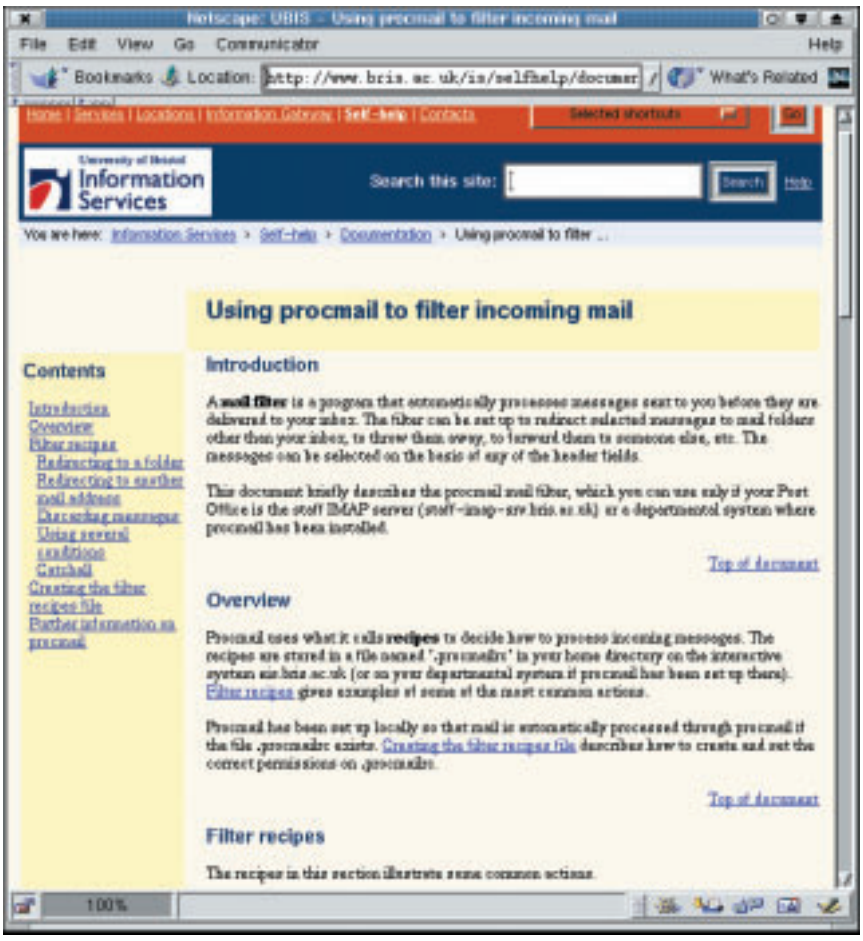
The .procmailrc file is nothing more than a shell script, there is nothing to stop us from calling other scripts should a filtering condition be met, and this is where the power and control lie. With some creative .procmailrc files you can automatically delete all these unwanted, unsolicited commercial emails which clog up the inbox, or fire klaxon warnings should you receive a message from your editor telling you how late your copy is. Procmail comes with some other utility programs – Formail is one. With these you could write a script to reformat digest mailing list messages back to their original form, making it much easier to follow the thread of a discussion, like so:

```
:0
* ^Subject:.*Digest
|formail +1 -d -s procmail

LOGFILE=$MAILDIR/from               # Put it➥
 here, in order to avoid logging
                                    # the➥
 arrival of the digest.
```

You're not limited to moving emails from folders – you can also copy them. This would allow you to keep logs of all emails received, recording just the details you think are necessary. This could mean merely logging the time, subject and From header fields or preserving the entire email, a copy of which could be sent directly to some archive/compression routine.

Procmail comes with a wealth of documentation and there are lots of third party tutorials, faqs and discussion groups probably because it is an ideal utility to tinker with, as well as providing a much-needed service. ■



Bristol University is helpful in setting up filters