

Mail Filtering with Procmail

Volker Kuhlmann
<help....@top.geek.nz>

23, 28 Sep 2004

Copyright © 2004 by Volker Kuhlmann

What can procmail do?

⇒ everything! 😊

- operate on a single email
- act on characteristics of the email
 - strings in headers or body
 - size
- actions
 - “deliver” email (write to mail spool)
 - send email to other recipient(s)
 - call other programs
- sort your mail
- eat your mail

How is procmail deployed?

- LDA/MDA
 - sendmail, postfix can easily be configured to use procmail for mail delivery
- called by fetchmail after POP3/IMAP download
- sorting of (e.g. IMAP) mail stores
- from the command line (e.g. pipe)
- ! procmail is not an MTA! !

Technicalities

- procmail reads the email from stdin
- the file containing the filter instructions can be
 - given on the command line
 - /etc/procmailrc
 - \$HOME/.procmailrc
- final location of mail depends on filter instructions, but is not usually stdout

Recipes

- filter instructions = recipes

```
:0 flags  
* condition  
action
```

- flags: affect the condition and action
- condition: an extended regular expression
 - some procmail-specific extensions
- action: store, pipe, mail, nesting block

Example: store CLUG mail to folder

```
:0 :  
* ^Comments: University of Canterbury Linux Users  
Mail/CLUG
```

- “.” = use locking
- regex
 - matches any header line
 - case-insensitive
- action: save mail to “Mail/CLUG” if all preceeding conditions were matched
- processing continues with next recipe as soon as a condition is false
- “falling off” delivers to system default

Flags

- some of the flags:
 - “:”: use a lockfile (required when storing to mbox)
 - “c”: start a copy (clone) of the mail + process
- with action “|”:
 - “f”: filter mail through program
 - “h”, “b”: write header, body, or both (default) to pipe
 - “w”: wait until the pipe terminates (needed with “f”)
 - “i”: ignore any write errors to pipe
(most likely caused because pipe doesn't read stdin)
- flags are case-sensitive

Conditions

- Suggested condition line order:

```
* $ ! ? BH VAR ?? test
```

- “\$”: expand variables
- “!”: negation
- “?”: return code (true/false) of program
- “BH”: egrep body, header, or both (default header)
- “VAR ?? regex”: match regex in variable VAR

Actions

- Storing
 - non-existing name or existing file: mbox file
 - existing directory: stored with some prefix
 - “name/”: maildir format
 - “name/.”: mh format
- Action: “|”
 - mail is piped into the given program
- Action: “!”
 - mail is forwarded to the given email addresses
- Action: “{ ... }”
 - nesting block

Other thoughts

- dealing with procmail trains understanding of how email works, how headers are used, etc
- it's easy to shoot yourself in the foot and create mail loops or annoy others
- issues of how to process mail are probably not specific to procmail
- change reply-to of list mails to go where you (not the list admin) want it

More Information

- <http://www.procmail.org/>,
procmail@lists.RWTH-Aachen.DE
- Era Eriksson's FAQ:
<http://www.iki.fi/era/procmail/mini-faq.html>
- Jari Aalto's Procmail resources:
<http://pm-doc.sourceforge.net/>
- Nancy McGough's Quick Start:
<http://www.ii.com/internet/robots/procmail/qs/>
- Volker's library:
<http://volker.top.geek.nz/soft/procmail/>
- `man procmail`, `procmailrc`, `procmailex`, `procmailsc`