

Procmail

Utilisation de procmail

Olivier Tharan

<olive@oban.frmug.org>

Ce document décrit l'utilisation de procmail dans la vie de tous les jours.

1. Introduction

Ce document essaie d'expliquer comment installer et utiliser Procmail pour filtrer le courrier électronique. Procmail est un outil très puissant pour traiter, de quelque manière que ce soit, des courriers électroniques. Ce document essaie de donner des exemples simples à mettre en oeuvre, tout en n'oubliant d'expliquer le pourquoi de ces exemples.

Les recettes sont tirées de mon expérience personnelle, et représentent mes commentaires sur les documentations largement disponibles.

NOTE : ce document est en cours d'élaboration et est évidemment très incomplet (mais il évolue peu à peu !). Les sources contiennent des commentaires guidant un peu l'évolution du document, les lire pour savoir où ajouter des éléments, chercher le mot 'FIXME' ! Comme il a été relu par peu d'autres personnes, il se peut que certaines informations soient erronées.

Ce document se trouve sur son site de référence
(<http://www.linux-france.org/article/appli/procmail.html>).

1.1. Qu'est-ce que procmail ?

Procmail est un outil permettant principalement de filtrer des messages électroniques (il ne faut pas oublier formail, livré avec procmail, qui est un adjoind appréciable à ce dernier). Il est très puissant et ses capacités de filtrage lui permettent de délivrer des courriers dans différentes boîtes aux lettres, de les renvoyer, voire d'effectuer n'importe quelle action en fonction du filtre désigné.

Procmail fonctionne grâce à un système de règles, qu'il parcourt les unes après les autres, afin de déterminer si le message qu'il traite satisfait à une règle en particulier, et exécute l'action associée à la règle trouvée, le cas échéant.

Vous avez peut-être entendu parler de procmail. Vous avez peut-être vu des règles procmail, et vous êtes enfui en courant après ça (ça se peut !). Vous connaissez la puissance de procmail, mais vous ne savez pas trop par où commencer. Ce document est pour vous. Il expliquera comment installer procmail, et comment l'utiliser de manière simple au début. Finalement, une série de pointeurs vous permettra d'aller plus loin, une fois les bases assimilées.

Pré-requis pour utiliser procmail :

- savoir utiliser le courrier électronique (je ne vais pas l'apprendre à votre place, mais si vous êtes là, c'est que vous savez, n'est-ce pas ? ;-)
- avoir des connaissances de base en expressions rationnelles (les fameuses regexp !), qui sont abondamment utilisées dans les règles de procmail. Rassurez-vous, j'essaierai de mettre suffisamment d'exemples pour que cela reste compréhensible. Sinon, procmail utilise les expressions rationnelles standards d'**egrep**, donc vous pouvez lire les pages de manuel `egrep(1)` et `regex(7)` pour plus d'informations.

2. Installation de procmail

2.1. Compilation

L'installation de procmail n'est pas très compliquée. Si vous n'êtes pas l'administrateur de votre système, il faudra contacter ce dernier pour savoir si procmail est installé ou pas sur le serveur de courrier. Si vous pouvez (ou devez !) l'installer vous-même, ce n'est pas difficile. En général, procmail est distribué sous forme d'un paquet pour les distributions Linux (`procmail-3.14-2.i386.rpm` pour la Redhat-6.2, `procmail_3.13.1-3.deb` pour la Debian 2.2, etc.) ou d'un port pour FreeBSD. Si vous ne trouvez pas de distribution binaire pour votre système, il est possible de le compiler à partir des sources, que l'on peut trouver sur le site de procmail (<ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail/procmail.tar.gz>) ou plus rapidement au Lip6 (Jussieu) (<ftp://ftp.lip6.fr/pub/unix/mail/procmail/procmail.tar.gz>). La dernière version à ce jour est 3.15.

2.2. Configuration

Il existe plusieurs manières d'appeler procmail, selon que l'administrateur système est coopératif ou non ! Il est en effet possible d'utiliser procmail de manière globale (procmail peut servir de filtre général, avant même la livraison du courrier à chaque utilisateur -- pour passer un anti-virus à l'entrée du système, par exemple), ou bien de manière individuelle (utilisation de procmail sans disposer de droits particuliers autres que ceux du répertoire personnel).

2.3. Installation personnelle

Il est possible d'installer procmail pour un usage personnel, si jamais le système n'est pas paramétré pour utiliser procmail conjointement avec le serveur de courrier. Pour cela, la documentation de procmail indique de mettre la ligne suivante dans le fichier `.forward` de votre répertoire personnel :

```
"| IFS=' '&& exec /usr/bin/procmail -f-||exit 75 #MON_LOGIN"
```

Notez le « tube » (|) en début de ligne ¹ ! Le fichier `.forward` est cherché par le serveur de courrier à chaque courrier qui nous est destiné ; si ce fichier existe, il permet de définir, soit une adresse à laquelle redistribuer (= forwarder) le courrier, soit une commande à travers laquelle on passe le courrier. Dans notre cas, on indique au serveur de courrier de passer les courriers entrants nous étant destinés à travers la commande `procmail`.

3. Interaction avec le reste du système

3.1. Procmail comme livreur de courrier

Une manière très utile de mettre `procmail` à disposition de tous les utilisateurs est de l'installer comme outil de livraison du courrier en local. `procmail` est en effet aussi un MDA (Mail Delivery Agent, outil de livraison de courrier), c'est-à-dire que le courrier local traité par le serveur de courrier est envoyé à `procmail` pour que ce dernier place le courrier dans les boîtes aux lettres des destinataires locaux (en général, dans `/var/mail/login`).

De manière historique, c'est l'outil `/bin/mail` qui faisait office de livreur de courrier local. Il a tendance à être remplacé par `procmail`, car ce dernier permet aussi de filtrer le courrier entrant. L'un des inconvénients de `procmail` est qu'il est plus lourd à lancer à chaque fois qu'un courrier doit être livré.

Par contre, une fois installé de manière globale, `procmail` peut servir de filtre pour les personnes qui le désirent, et comme simple livreur de courrier pour les autres. Il suffit pour les premiers d'avoir un fichier de configuration `.procmailrc` dans leur répertoire personnel. Si ce fichier n'est pas présent, `procmail` livrera le courrier dans la boîte aux lettres habituelle de la personne.

L'installation de procmail comme MDA local est relativement simple pour les serveurs de courrier les plus courants. Étudions les cas les plus courants.

3.2. Procmail et Sendmail

Si vous paramétrez sendmail avec les macros m4, il suffit de placer la ligne suivante dans `sendmail.mc` (ou `config.mc` suivant l'Unix utilisé ²) :

```
FEATURE(local_procmail)dnl
```

et, si votre procmail n'est pas dans `/usr/local/bin` ³ :

```
define('PROCMAIL_MAILER_PATH', '/usr/bin/procmail')dnl
```

3.3. Procmail et Postfix

Pour utiliser procmail avec Postfix, rien de plus simple : renseignez la ligne `mailbox_command` du fichier `main.cf` ⁴, en indiquant le chemin complet vers procmail. Par exemple :

```
mailbox_command = /usr/bin/procmail
```

3.4. Exim

Pour utiliser procmail avec Exim, il faut remplacer le contenu habituel du `~/ .forward` par :

```
|/usr/bin/procmail
```

Cette solution devrait en fait fonctionner avec tout serveur de courrier possédant une compatibilité proche de Sendmail. Si quelqu'un a une solution générique ne nécessitant pas l'écriture d'un fichier `~/ .forward`, elle est la bienvenue !

3.5. Autres serveurs de courrier

FIXME: qui a envie d'écrire là-dessus ? QMail, etc.

3.6. Procmail et fetchmail

Ceux qui se connectent épisodiquement à Internet par une liaison PPP utilisent peut-être fetchmail pour récupérer leur courrier depuis le serveur POP ou IMAP de leur fournisseur d'accès. fetchmail est un très bon client POP/IMAP, que je vous recommande.

En temps normal, fetchmail récupère le courrier sur le serveur distant et l'adresse au serveur de courrier local, sur le port 25 (SMTP). Cette méthode fonctionne si vous avez configuré un serveur de courrier sur votre machine (Sendmail ou Postfix), mais est relativement lourde car une fois passé à Sendmail, le courrier finira par passer dans les mains de procmail. On peut éviter un saut supplémentaire en faisant passer le courrier directement de fetchmail à procmail. Pour cela, il suffit d'ajouter l'option de configuration suivante dans le fichier `.fetchmailrc` :

```
mda "/usr/bin/procmail -Y -d %T"
```

Le paramètre `-mda` passé en ligne de commande fait la même chose.

3.7. Configurer Gnus pour qu'il utilise procmail

Éric Jacoboni a contribué cette section sur l'utilisation de Gnus conjointement à Procmail. Gnus est suffisamment important et puissant pour qu'on doive le configurer

de manière à l'utiliser proprement avec procmail.

Gnus permet de créer de façon simple des dossiers de courrier selon les critères de votre choix : il suffit d'utiliser la fonction `nnmail-split-methods` et de spécifier, pour chaque dossier, l'expression rationnelle sur laquelle on se basera pour classer un courrier dans celui-ci.

Voici, par exemple, ce que pourrait contenir un `.gnus` pour activer cette possibilité :

```
;; On utilise le backend nnml pour le courrier (car par défaut,
;; Gnus lit les news...)
(setq
  gnus-secondary-select-methods '((nnml ""))
  mail-sources
    '((directory
      :path "~/incoming/" ;; cf. variable MAILDIR de .procmailrc
      :suffix "" )
    )
)
```

Ici, nous avons choisi d'utiliser le répertoire `incoming` comme source de courrier, c'est celui où procmail range ses dossiers. Gnus transférera alors les courriers dans les répertoires correspondants de Mail (c'est le répertoire par défaut).

Toutefois, cette méthode a quelques inconvénients : le principal est qu'elle fait double emploi si l'on utilise déjà procmail pour, par exemple, mettre aux oubliettes des courriers de provenance douteuse, ou pour supprimer les doublons (voir précédemment). Un autre inconvénient est qu'elle est propre à Gnus, ce qui rend obsolètes tous ces dossiers si l'on décide de changer de MUA (mais quelle idée saugrenue...).

Pour cette raison, il est plus sage de configurer Gnus pour qu'il utilise procmail pour gérer ces dossiers :

```
;; Utilisation du backend nnml avec procmail
;; nnmail-procmail-directory correspond à la variable MAILDIR
;; de votre .procmailrc
(setq
  gnus-secondary-select-methods '((nnml ""))
```

Procmail

```
gnus-use-procmail t
nnmail-spool-file "procmail"
nnmail-procmail-directory "~/incoming/"
nnmail-procmail-suffix ""
)

;; Pour les nouvelles versions de Gnus (>= 5.8.7)
(setq mail-sources
  (list '(directory
         :path "~/incoming/"
         :suffix "")
        )
  )
)
```

et c'est tout... Gnus utilisera les règles définies dans votre `.procmailrc` pour créer ses dossiers de courrier.

Le principe est simple : lorsque vous lancerez Gnus, celui-ci ira consulter `~/incoming` où doivent se trouver les différents fichiers créés par procmail. Si ces fichiers ne sont pas vides, il les lit et transfère les courriers qu'ils contiennent dans les sous-répertoires de `~/Mail` en créant un fichier par courrier. Après cette étape, les fichiers de `~/incoming` sont donc tous vides.

Si vous avez, par exemple, les règles suivantes dans votre `.procmailrc` :

```
# *FreeBSD-Stable
:0
* ^TO.*stable@FreeBSD
FreeBSD-Stable

# *Systeme
:0
* From:.*(root|news)@chezmoi
Systeme

# *Perso
:0
* ^TO.*moi
Perso
```

```
# *Le-reste
:0
* ^TO.*
Le-Reste
```

Vous aurez les fichiers `~/incoming/FreeBSD-Stable`, `~/incoming/Systeme`, `~/incoming/Perso` et `~/incoming/Le-Reste` qui se rempliront au fur et à mesure que vous recevez des messages correspondant aux règles ci-dessus. En lançant Gnus, celui-ci créera (si ce n'est déjà fait) les répertoires `~/Mail/FreeBSD-Stable`, `~/Mail/Systeme`, `~/Mail/Perso` et `~/Mail/Le-Reste` et chacun d'eux contiendra des fichiers portant des noms sur le même principe qu'un spool de news :

```
% ls -l ~/incoming/Perso
-rw----- 1 moi moi 0 26 jul 14:05 /home/moi/incoming/Perso
          ^^^

% ls ~/Mail/Perso
1      11      13      2      4      6      8
10     12     14      3      5      7      9
```

L'avantage de cette méthode est de centraliser le triage du courrier en un seul endroit : `.procmailrc`. D'autre part, procmail dispose de moyens bien plus perfectionnés pour spécifier les expressions rationnelles (la macro `^TO_`, notamment).

Par contre, Gnus considèrera que chaque fichier de `~/incoming` contient une liste de courriers : si vous avez donc créé un fichier `msgid.cache` dans ce répertoire avec la règle

```
:0 Wh: msgid.lock
| formail -D 8192 msgid.cache
```

vue plus haut, Gnus se plaindra que ce dossier n'est pas au bon format. Pour régler ce problème, il suffit de modifier cette entrée en :

```
:0 Wh: msgid.lock
```

```
| formail -D 8192 /tmp/msgid.cache
```

ce qui a l'avantage supplémentaire de nettoyer ce fichier à chaque redémarrage de votre système (s'il est configuré pour nettoyer /tmp).

Ressources :

Comme d'habitude, le fichier info de Gnus donne tous les renseignements (Rubrique 'Select Methods' -> 'Getting Mail' -> 'Mail and Procmail').

La Faq Gnus (<http://www.ccs.neu.edu/software/contrib/gnus/>) contient également une rubrique sur cette configuration (contribution de Randal Schwartz).

Gnus RuLeZ !

3.8. Utilisation de Mutt avec Procmail

Mutt s'accommode très bien à la sauce procmail. Il lit les boîtes au format mbox, format par défaut sous Unix, qui est le format standard écrit par procmail. Il suffit donc d'indiquer à mutt le répertoire qu'utilise procmail pour stocker les boîtes aux lettres. Si vous utilisez la configuration indiquée ci-dessous, le répertoire est \$HOME/Mail, qui est le répertoire par défaut utilisé par mutt. La boîte aux lettres normale, celle où procmail stockera le courrier qui n'a pas été trié et que mutt lira au démarrage sans autre indication de votre part, est /var/mail/\$LOGIN (ou /var/spool/mail/\$LOGIN).

Si vous changez le répertoire où procmail stockera le courrier, il faut aussi l'indiquer à mutt, en utilisant la variable `folder`. Par exemple : `set folder=~courrier` dans votre `.muttrc`. Voir la variable `MAILDIR` de procmail.

Si vous modifiez la boîte aux lettres système (celle où devraient être stockés vos courriers en définitive) avec la variable `DEFAULT` de procmail, il faudra aussi l'indiquer à mutt. Pour cela, vous pouvez soit utiliser la variable `spoolfile` dans la configuration de mutt : `set spoolfile=~Mail/inbox`, soit positionner la variable d'environnement `MAIL`.

Tous ces renseignements se trouvent dans le manuel de mutt (installé en général dans /usr/share/doc/mutt/manual.txt).

4. Configuration de procmail

L'utilisation de procmail revient à peu près à se créer un fichier de configuration ; c'est peut-être la partie la plus difficile du parcours, mais avec des exemples bien choisis, c'est plus clair.

4.1. Écriture du fichier `.procmailrc`

Le fichier `.procmailrc` est une combinaison d'assignations de variables et de *recettes*. Les recettes procmail sont composées de trois parties :

- l'en-tête et ses options ;
- les règles de correspondance : expressions rationnelles sur les en-têtes et corps des messages permettant de déterminer si le message va être traité par cette recette ;
- l'action associée à cette recette.

La première correspondance trouvée l'emporte, et termine la lecture de `.procmailrc` ; si aucune correspondance n'est trouvée, le courrier est livré dans la boîte aux lettres par défaut (`/var/mail/toto` par exemple).

4.1.1. Variables

Il est possible de définir des variables dans le fichier de configuration `.procmailrc`. Certaines d'entre elles auront une signification particulière pour procmail, alors que les autres peuvent être utilisées dans les règles qui suivront comme des variables normales. Voici un exemple commenté :

```
# Variables spéciales pour procmail

# caractère verbeux de procmail ; mettre 'yes' per-
# met d'avoir des messages
# supplémentaires
VERBOSE=no
```

```
# pas obligatoire : procmail détermine votre nom de lo-
gin tout seul, mais pour
# l'exemple on supposera que l'utilisateur s'appelle 'toto'
LOGNAME=toto

# mettre /bin/sh surtout si vous utilisez tcsh !
SHELL=/bin/sh

# chemin d'accès aux exécutable ; en mettre le mini-
mum, pour n'accéder qu'aux
# programmes indiqués dans le fichier de configuration
PATH=/bin:/usr/bin:/usr/local/bin:/home/toto/bin

# réper-
toire où seront stockés les mails ; s'assurer que votre MUA sait y
# accéder aussi
MAILDIR=/home/toto/Mail

# si procmail n'arrive pas à délivrer le cour-
rier, cette boîte sera utilisée
# en dernier ressort : il vaut mieux définir cette variable !
ORGMAIL=$MAILDIR/emergency-inbox

# boîte de réception par défaut
DEFAULT=/var/mail/toto

# fichier de log de procmail ; si vous définissez cette variable,
# procmail gardera une trace de son exécution dans le fichier
# indiqué. À consulter périodiquement !
LOGFILE=$MAILDIR/.procmail.log
```

Les autres variables utilisées ou définies par procmail sont décrites dans la page de manuel de `procmailrc(5)`.

4.1.2. Séparer le fichier en plusieurs parties

Il est possible de scinder le fichier `.procmailrc` en plusieurs parties, afin d'en améliorer la lisibilité. En fait, on peut inclure un fichier grâce à la directive suivante :

```
INCLUDERC=nom_de_fichier
```

où `nom_de_fichier` est le fichier à inclure à l'endroit de la directive, c'est soit un chemin complet, soit un chemin relatif au répertoire courant. On peut inclure des fichiers de façon imbriquée sans aucun problème.

4.1.3. Recettes

Une recette a le format suivant :

```
:0 [drapeaux] [ : [verrou_local] ]
<zéro ou plusieurs conditions (une par ligne)>
<exactement une ligne d'action>
```

Une recette commence par `:0` ; sur la même ligne, on ajoutera des drapeaux supplémentaires selon les cas. Les drapeaux sont décrits dans la page de manuel de `procmailrc(1)`. Les plus utilisés sont :

- `H` : passe l'expression rationnelle sur les en-têtes du message ;
- `B` : idem sur le corps ;
- `h` : n'envoie que les en-têtes pour traitement ;
- `c` : garde une copie du message pour traitement ultérieur

Finalement, en fin de ligne, on ajoutera `:` si on désire un fichier de verrou (afin de ne pas corrompre un fichier en y écrivant simultanément deux courriers).

Les lignes suivantes indiquent zéro ou plusieurs conditions, chacune sur une ligne différente, permettant de tester si un courrier correspond à ce que l'on cherche. Les conditions commencent par le caractère `*` et tout ce qui suit est envoyé à la commande **egrep** interne à procmail. Il faut que toutes les conditions soient vraies pour considérer la recette comme vraie et activer l'action qui suit (conditions de type AND).

Finalement, l'action à effectuer si les conditions ont été vérifiées sont :

- la redirection vers un fichier : action la plus courante, elle permet de délivrer le courrier dans la boîte aux lettres indiquée. Il faudra faire attention à spécifier un verrou au début de la recette. Il suffit d'indiquer sur la ligne le nom du fichier ou du répertoire dans lequel stocker le courrier ;
- la redirection vers un programme : cette action permet d'envoyer le courrier sur l'entrée standard d'un programme qui s'occupera du courrier. Le verrou n'est plus nécessaire, et on indiquera la redirection en commençant la ligne par une barre verticale (|) puis la commande à exécuter ;
- l'envoi vers une autre adresse électronique : le verrou est inutile, et la ligne d'action commencera par un point d'exclamation, suivi de l'adresse de renvoi.

4.2. Quelques recettes utiles

Voici une liste désordonnée de recettes de base qui pourront être utiles, et qui sont différentes des règles qu'on trouve dans les pages de manuel de procmail.

1. Ne garder qu'un seul exemplaire d'un mail, même si on le reçoit deux fois ; utile, par exemple, quand on reçoit une réponse d'une liste de diffusion, à la fois directement et par l'intermédiaire de la liste. Il est possible de régler la taille du cache en changeant le nombre (8192).

```
:0 Wh: msgid.lock
| formail -D 8192 msgid.cache
```

2. Ces règles permettent de trier une liste de diffusion dans un dossier séparé, afin de ne pas trop encombrer la boîte aux lettres normale.

Imaginons que je sois abonné à la liste Postfix-Fr. L'adresse pour envoyer un message à cette liste est <postfix-fr@lists.freenix.org>, mais quand vous en recevez, l'adresse de l'expéditeur est <postfix-fr-owner@lists.freenix.org>. Il nous faut donc trouver une

expression suffisamment complète qui nous permette de repérer un tel message sans ambiguïté. Une telle expression peut être exprimée par la règle suivante :

```
:0:
* ^From.*postfix-fr-owner
postfix-fr
```

Dans cette règle, nous recherchons si l'en-tête représentant l'expéditeur (la partie `^From` de l'expression régulière) contient la chaîne `postfix-fr-owner`, et si c'est le cas, plaçons le courrier dans la boîte aux lettres nommée « `postfix-fr` ».

Pour trier d'autres listes de diffusion, le principe est le même : trouvez dans les entêtes des courriers des éléments suffisamment significatifs sur lesquels vous baserez votre tri (`From.*liste-owner` pour une liste Sympa, `From.*owner-liste` pour une liste Majordomo, `From.*liste-admin` pour une liste Mailman, etc.).

3. La règle qui suit permet de trier un courrier sur son sujet, ce qui permet de filtrer certaines listes de diffusion ajoutant systématiquement leur nom dans le sujet du courrier. Vous pouvez aussi discuter d'un thème précis avec des amis, des collègues en convenant d'un mot-clé dans le sujet.

```
:0:
* ^Subject:.*\[blague\]
copains/blagues
```

Tous les courriers dont le sujet contient la chaîne `[blague]` (avec les crochets) iront dans le dossier `blagues` se trouvant dans le sous-répertoire `copains` de votre répertoire de courrier. Les crochets sont échappés par une barre oblique inverse (`\`) car il s'agit d'une expression rationnelle dans laquelle les crochets ont une signification particulière.

4. La règle suivante, assez impressionnante, permet de trier en un seul coup tous les courriers arrivant de listes de diffusion similaires. L'exemple typique en est les

listes de diffusion du projet Debian (<http://www.debian.org/>), auxquelles la règle fait référence (mais cela pourrait fonctionner aussi pour les listes FreeBSD ;-) :

```
:0 H
* ^X-Mailing-List:.*[<].*lists\.debian\.org[>]
* ^X-Mailing-List:.*[<] *\^[^ ]*[@]*
$MATCH
```

Cette règle à l'apparence barbare s'appuie sur le fait que les courriers des listes Debian contiennent l'entête X-Mailing-List :
debian-nom-liste@debian.org, et que la variable \$MATCH contient ce qui a été trouvé dans l'expression rationnelle précédente à partir de \/. Pour clarifier, l'expression \^[^]*[@]* « correspondra » à un début d'adresse électronique (sans la partie domaine) et sera affecté à la variable \$MATCH. Pour la liste debian-french@lists.debian.org par exemple, le courrier donc sera placé dans le dossier debian-french.

Cette méthode fonctionne pour toutes les listes Debian, si vous êtes comme moi abonné à plusieurs d'entre elles.

5. Question : je récupère mon courrier depuis une seule adresse, et j'aimerais le trier en fonction du destinataire, selon que l'adresse soit Jean Dupont <toto@fai.fr> ou Suzanne Dupont <toto@fai.fr>.

Une solution peut être d'utiliser fetchmail et son mode *multidrop*, ou bien les règles procmail suivantes :

```
:0
* ^TO_Suzanne.*Dupont
! suzanne

:0
* ^TO_Jean.*Dupont
! jean
```

Faites toutefois attention à ce qu'une boîte aux lettres récupère le courrier qui serait éventuellement passé à travers ; ce sera le cas quand l'adresse du destinataire

n'aura pas la forme souhaitée. C'est en tout cas une façon rapide, mais peu efficace, de distribuer du courrier en provenance d'une seule adresse à destination de plusieurs personnes. Une solution plus pratique, par exemple, est d'utiliser UUCP.

6. L'utilisation la plus attendue de procmail est le traitement des spams. Bien qu'il soit possible de les bloquer plus en amont (listes noires avec les serveurs de courriers, comme RSS, ORBS, MAPS, etc.), on déteste généralement les voir arriver dans la boîte aux lettres. Je vous renvoie sur les sites *Cauce* (<http://www.cauce.org/>) pour les raisons philosophiques de la lutte contre le spam et propose une solution simple pour mettre de côté les courriers non désirés. Je n'indique pas comment les jeter (`/dev/null`) car j'ai eu de mauvaises surprises : mieux vaut avoir un ou deux spams dans la boîte qu'un courrier important à la trappe !

La règle qui suit utilise la méthode de la « liste noire », que l'on gère en insérant dans un fichier une adresse non désirée, voire une expression rationnelle si l'on veut bloquer un domaine entier, par exemple. Elle utilise deux programmes, `extract-addr` et `match-email-addr` tirés de la définition de cette règle adaptée de liste noire. Pour l'historique, je me suis basé sur les règles de Larz Wirzenius (fameux ancien modérateur de `comp.os.linux.announce`) disponibles à <http://www.iki.fi/liw/mailfilter.html> ; Richard Gooch (fameux « hacker » du noyau Linux, entre autres) a adouci ces règles et met les sources des deux programmes cités à disposition à <ftp://ftp.atnf.csiro.au/pub/people/rgooch/>⁵.

Compilez ces deux programmes, mettez-les à un endroit accessible par procmail (rappelez-vous de la variable `PATH` du `.procmailrc`), et incluez la règle suivante :

```
# À modifier selon les cas
BLACKLIST=$MAILDIR/.blacklist

:0 Whic
| extract-addr | match-email-addr $BLACKLIST

# Si on a trouvé une adresse de spammeur (ci-dessus), alors...
:0 a
{
:0:
```

```
junk/spam  
}
```

Mettez de préférence cette règle vers la fin de votre fichier de configuration procmail, car elle passe le courrier à travers deux tubes et est un peu plus gourmande que les autres vues auparavant. Les deux premières lignes envoient le courrier sur `extract-addr` qui extrait les adresses électroniques du message et les envoie à `match-email-addr` qui les comparera avec le contenu de la liste noire.

En cas de correspondance, le résultat de la commande sera positif et la règle suivante sera exécutée, ce qui a basiquement pour effet de mettre le courrier de côté. Lisez cette boîte de temps en temps, on ne sait jamais...

Le contenu du fichier de liste noire est le suivant. Je vous conseille vivement de créer vous-même votre fichier de liste noire, au fur et à mesure des nouveaux spams. Couplée à d'autres règles (pas de HTML, est-ce que le courrier m'est bien adressé, etc.), cette méthode réduit énormément le nombre de spams dans la boîte principale (mais pas le nombre absolu de courriers non sollicités, hélas).

```
# Les commentaires ainsi sont possibles --  
en tout cas, ça marche chez moi  
# (tm) -- voir le source de match-email-  
addr pour plus de détails  
  
# quelques adresses « connues »  
Friend@public.com  
friend@everywhere.com  
BeSeen@At.Our.Site.com  
  
# quelques expressions rationnelles ; je suis un peu bour-  
rin sur ce coup  
# le mot-clé !regexp est important  
!regexp [@.]msn\.com  
!regexp [@.]jaol\.com  
!regexp [.]writeme\.com
```

Note : ce passage sur le spam devrait faire l'objet d'une section complète pour la prochaine révision du document.

4.3. Utilisation en dehors d'un serveur de courrier

4.3.1. Formail

Pour retraiter une boîte aux lettres non triée avec un filtre particulier, on peut utiliser `formail`, qui appellera `procmail` à son tour. Exemple pratique :

```
formail -s procmail fichier_filtre < boite_a_traiter
```

où `fichier_filtre` est un fichier contenant les règles `procmail` nécessaires au traitement du fichier et `boite_a_traiter` est la boîte aux lettres contenant les courriers à filtrer.

On peut omettre `fichier_filtre`, auquel cas le fichier de configuration par défaut (`.procmailrc`) sera utilisé.

`Formail` permet bien d'autres choses utiles, notamment celle d'extraire des champs particuliers des entêtes d'un courrier, ou d'ajouter des entêtes particuliers. Ceci peut être utile dans le cas d'un répondeur automatique (explication décortiquée dans la section sur l'utilisation avancée de `procmail` ci-dessous).

L'utilisation de `formail` est documentée dans la page de manuel `formail(1)`.

4.3.2. Mailstat

Si vous générez des traces de l'utilisation de `procmail` (par l'utilisation de la variable `LOGFILE`, par exemple), vous aurez envie d'en avoir un condensé régulier ; les lire exhaustivement peut prendre du temps. L'outil « `mailstat` », livré avec `Procmail`, peut servir à cela ⁶.

La commande simple suivante permet de générer un résumé des traces de `procmail` tout en gardant le fichier de trace intact :

```
mailstat -klm procmail.log
```

Le script suivant permet une utilisation plus avancée, à lancer régulièrement (une fois par semaine, par exemple) :

```
#!/bin/sh
# zerostat : archive les logs de procmail, envoie un ré-
# sumé par email

# Variables à modifier en fonction de votre environnement

# mettre votre login
MOI=toto

# indiquer le fichier de trace
LOGFILE=~/.Mail/.procmail.log

# utilisez gzip ou bzip2, au choix
ZIPEXE=bzip2

PATH=/bin:/usr/bin:/usr/local/bin

mailstat -l $LOGFILE | \
mail -s 'Statistiques procmail' $MOI
$ZIPEXE -f ${LOGFILE}.old
```

L'utilisation de mailstat est documentée (succinctement) dans la page de manuel mailstat(1). Ne pas oublier de passer le nom du fichier à traiter en paramètre !

5. Utilisation avancée

On peut utiliser procmail pour d'autres utilisations avancées, notamment en tant que filtre global pour traiter tout le courrier entrant. Les utilisations possibles sont de pouvoir filtrer les spams, les virus, faire des actions définies, avant que le courrier ne soit délivré à chaque utilisateur. Ceci peut être très pratique sur un relais de courrier à l'entrée d'une organisation par exemple.

Une autre utilisation possible est de mettre en place un répondeur automatique. Cela peut servir à plusieurs choses.

- Mettre un répondeur quand on part en vacances et qu'on ne répondra pas tout de suite, mais en signalant à l'expéditeur que le message est conservé (principe du programme vacation).

Problèmes posés : ne pas répondre deux fois à la même personne (utilisation d'un hachage pour repérer les personnes auxquelles le filtre a déjà répondu), ne pas répondre aux listes de diffusion (c'est très ennuyeux, on supposera que les listes sont déjà triées quand le message arrive, on vérifiera quand même qu'on ne tombe pas sur un gestionnaire de listes, etc.).

- Faire un mini-serveur de fichiers, déclenché par une balise spéciale dans le sujet (exemple : Subject: GET liste-fichiers.gz)

Problèmes posés : se limiter à une liste de fichier déterminée, ne pas accepter les caractères bizarres qui pourraient créer un trou de sécurité et permettre de servir n'importe quel fichier du système.

5.1. Refuser des attachements (Word, HTML, etc.)

Thomas Nemeth a donné la recette qu'il utilise pour filtrer des messages contenant des attachements indésirables : scripts Visual Basic (.vbs), des exécutable Windows (.exe), des documents Word (.doc) ou Excel (.xls). Cette recette a pour effet de renvoyer un message à l'expéditeur lui demandant de ne pas envoyer ce genre d'attachements.

```
:0 HB
* ^Content-Type:.*\.(exe|vbs|doc|dot|xls)
*$ !X-Loop:.*$MOI
| (formail -rtk -p ' | ' \
  -A "X-Loop: $MOI" \
  -A "Precedence: junk"; \
  cat $HOME/.procmail/badfile ) | $SENDMAIL -t -oi
```

Cette recette utilise principalement formail pour générer un message d'auto-réponse. On recherche, à la fois dans les en-têtes et dans le corps (drapeaux 'H' et 'B'), un en-tête MIME décrivant le contenu (`Content-Type`) et indiquant un nom de fichier se terminant par les extensions définies dans l'expression (`exe|vbs|doc|dot|xls`). Libre à vous d'en ajouter ou d'en retirer.

En cas de correspondance, on génère une réponse automatique, qui inclura le message complet : celui-ci est donc perdu pour un traitement ultérieur, copiez-le si vous voulez le garder quand même. Les options utilisées sont décrites dans la page de manuel de formail. On inclut le message d'origine, puis un message expliquant le refus et le tout est passé à Sendmail.

Vous avez sûrement remarqué la troisième ligne de cette recette. Expliquons-la plus en détail. Quand nous générons le message, nous ajoutons un en-tête `X-Loop` contenant notre adresse électronique (la variable `$MOI` est à renseigner au début du fichier de configuration), qui servira à procmail pour tester si le message est déjà passé entre ses mains. Ceci a un but : éviter les boucles de courrier, qui peuvent s'avérer très gênantes à découvrir et à réparer. On teste donc avec cette règle que le message n'a pas déjà subi un filtrage. Le signe `$` derrière l'étoile en début de ligne indique à procmail que se trouvent dans la ligne des variables qu'il faudra substituer (ici, `$MOI`).

Il est possible d'optimiser cette recette, pour coller plus spécifiquement au format MIME. En effet, l'en-tête `Content-Type` contient le type MIME de l'attachement, qui est plus donc plus fiable à chercher que l'extension du nom de fichier. On pourrait écrire la deuxième ligne ainsi, pour chercher les attachements binaires Windows, par exemple :

```
* ^Content-Type:.*application/octet-stream.*name=.*\.exe
```

L'expression rationnelle cherche les en-têtes représentant un attachement de type `application/octet-stream`, dont le nom se termine par `.exe`. On peut faire de même avec d'autres types d'attachement, le mieux est de tester. Le tableau suivant

représente des attachements courants avec leur type MIME correspondant :

Table 1. Attachements et types MIME

| type MIME | extension | nom courant |
|--|------------------|-------------------------|
| application/octet-stream | .exe | binaire Windows |
| application/msword | .doc | document Word |
| application/vnd.ms-excel | .xls | document Excel |
| application/vnd.ms-powerpoint | .pps, .ppt | document PowerPoint |
| application/x-zip-compressed | .zip | document compressé Zip |
| application/x-unknown-content-type-hlpfile | .hlp | document d'aide Windows |
| application/pdf | .pdf | document PDF |
| image/jpeg | .jpg, .jpeg | image JPEG |
| image/gif | .gif | image GIF |
| text/html | .htm, .html | document HTML |
| text/x-vcard | | VCard Netscape |

6. Conclusion

6.1. Quelques ressources en ligne

Je vais tenter de résumer les principales ressources et documentations disponibles sur

procmail. Il n'existe pas à ma connaissance de livre sur procmail : une recherche sur la chaîne `procmail` sur le site Amazon (GB) (<http://www.amazon.co.uk/>) n'a rien donné, ni sur le site Amazon (USA) (<http://www.amazon.com/>). Par contre, des références à procmail existent dans les livres publiés chez O'Reilly.

Je tiens d'ailleurs à noter que je ne souhaite pas faire un repompage ou une traduction des documents que j'énumère ici, sinon ce ne serait pas du jeu. Si je trouve des morceaux intéressants sur le forum `fr.comp.mail` ou que quelqu'un m'envoie des informations intéressantes, je les inclue dans ce document.

Voici une liste, non exhaustive évidemment, de sites traitant principalement de procmail.

6.1.1. Sites en anglais

- Procmail.org (<http://www.procmail.org/>)
- FAQ sur procmail (<http://www.iki.fi/era/procmail/mini-faq.html>)
- Ressources sur Procmail (<http://www.procmail.org/jari/>)

6.1.2. Sites en français

- un bon article sur procmail se trouve dans le numéro 26 de la Linux Gazette : Linux Gazette en français (<http://www.linux-france.org/article/lgazette/issue-26/issue-26-4.html>)
- ce document se trouve sur son site de référence : Linux-France (<http://www.linux-france.org/article/appli/procmail.html>)

6.2. Remerciements

- Nat Makarévitch, pour l'hébergement sur Linux-France ;
- Stéphane Écolivet, mainteneur de la section applications sur Linux-France, qui m'a poussé à faire ce document quand il n'était qu'en gestation dans ma tête, et qui m'a fourni les premiers patchs ;
- Éric Jacoboni, qui a écrit la section sur l'utilisation de Gnus avec Procmail ;
- Thomas Nemeth, qui a donné la recette pour filtrer les attachements, ainsi que d'autres éléments ;
- Nicolas Le Scouarnec, Didier Belot, contributeurs de nombreuses astuces ;
- les contributeurs du forum `fr.comp.mail` ;
- tous ceux que j'ai oubliés, évidemment...

Notes

1. et toutes mes excuses pour l'avoir oublié au début...
2. Ce fichier est généralement situé dans `/etc/mail` ou dans `/etc/sendmail`, voire, d'après Éric Jacoboni dans l'article Configuration simple (ou simplifiée) de sendmail pour PPP (<http://www.linux-france.org/article/mail/sendmail/sendmail.html>), dans `/usr/lib/sendmail-cf/cf` si le système utilisé est une distribution RedHat de Linux.
3. `/usr/local/bin/procmail` est l'emplacement par défaut de **procmail** selon **sendmail**.
4. Ce fichier se trouve en général dans `/etc/postfix`.
5. Le parcours des règles de ces deux personnes est très instructif et témoignent d'une utilisation avancée de procmail.

Procmail

6. Ne pas confondre avec « mailstats » (notez le 's'), utilisé avec Sendmail !

