Administration du courrier électronique sous Linux

Auteur : Guylhem Aznar <guylhem at oeil.qc.ca>.
Traducteurs : Gacquer Frédéric <gacquer at neuronnexion.fr> et Dimitri Ara <dimitri.ara at mail.dotcom.fr>.
v3.2, Janvier 2000

Ce document décrit la mise en place, l'administration et la distribution du courrier électronique sous Linux. Il est destiné aux administrateurs plutôt qu'aux utilisateurs. (Voyez le *Mail-User-HOWTO* pour des informations sur la gestion du courrier côté utilisateur et sur les logiciels de courrier électronique.) Vous avez besoin de lire ce HOWTO si vous prévoyez de communiquer localement ou vers des sites distants via courrier électronique. Vous n'avez probablement *pas* besoin de lire ce document si vous n'échangez pas de courriers électroniques avec d'autres personnes sur votre système ou avec d'autres sites.

Contents

1 Introduction, copyright et mise en garde habituelle

1.1 Courrier électronique et courrier non sollicité (*spamming*)

Pour envoyer un courrier électronique à une personne mentionnée dans ce document, remplacer le *at* dans l'adresse électronique par un @.

Cette conversion est simple pour un humain, mais pas pour un outil automatique de récolte d'adresses ; c'est par conséquent utile pour protéger nos généreux contributeurs des courriers non sollicités.

1.2 Objectifs

Le but de ce document est de répondre à quelques-uns des commentaires et des questions qui semblent correspondre aux « questions fréquemment posées » (FAQ) sur les logiciels de courrier électronique sous Linux en général et aux versions des distributions RedHat et Debian en particulier.

1.3 Nouvelles versions

Des nouvelles versions (NDT : en anglais) de ce document seront postées régulièrement sur *comp.os.linux.announce*, *comp.answers* et *mail.answers*. Elles seront également ajoutées aux différents sites ftp anonymes qui archivent de tels documents comme par exemple *sunsite.unc.edu* `ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO` .

De plus, vous pourrez trouver ce document sur la page du *Linux Documentation Project* `http://sunsite.unc.edu/mdw/index.html` .

1.4 Réactions

Je suis intéressé par toute réaction, qu'elle soit positive ou négative, à propos du contenu de ce document. N'hésitez pas à me contacter si vous trouvez des erreurs ou de sérieux oublis.

Je lis mais ne réponds pas forcément à tous le courrier électronique que je reçois. Les suggestions d'améliorations seront examinées et traitées en fonction de mon temps, du bien fondé de la requête et de ma pression artérielle :-)

Les remarques désobligeantes et les courriers incendiaires seront expédiées vers `/dev/null`, ce n'est donc pas la peine de vous casser la tête.

Les remarques concernant le format actuel de ce document doivent être adressées au coordinateur des HOWTO : Tim Bynum (<`howto at wallybox.cei.net`>).

1.5 Copyright

1.6 Garantie limitée

Bien sûr, je dénie toute responsabilité quant au contenu de ce document. L'utilisation des concepts, des exemples et de tout autre contenu de ce document est entièrement vos propres risques.

2 Autres sources d'information

2.1 *Mail-User-HOWTO*

Il existe un *Mail-User-HOWTO* qui aborde le côté utilisateur du courrier électronique.

Il est maintenu par Eric S. Raymond ; vous pouvez le trouver sur *metalab.unc.edu* `http://metalab.unc.edu/LDP/HOWTO/Mail-User-HOWTO.html` .

2.2 USENET

Il n'y a rien de spécifique à Linux (ce qui n'a pas toujours été le cas) dans la configuration et l'utilisation de logiciels de messagerie. De ce fait, vous n'aurez certainement jamais à poster de questions relatives au courrier électronique dans le forum de discussion général *fr.comp.os.linux.configuration*.

Ne postez dans la hiérarchie fr.comp.os.linux que si votre question est vraiment spécifique à Linux comme par exemple « Avec quelles options le binaire sendmail de la Debian 1.2 est-il compilé ? » ou « Le smail 5.0 de RedHat plante quand je le lance ».

Laissez-moi le redire.

Il n'y a plus aucune raison pour poster une question sur les courriers électroniques dans la hiérarchie fr.comp.os.linux. Poster dans fr.comp.mail qui est le forum adéquat. Vous y trouverez des réponses à *toutes* vos questions.

Si vous postez dans fr.comp.os.linux. pour une question non spécifique à Linux, vous êtes au mauvais endroit. Les experts du courrier électronique traînent sur les forums indiqués ci-dessus et ne n'utilisent généralement pas Linux. Poster dans la hiérarchie Linux une question non spécifique à Linux est une perte de temps pour tout le monde.*

En plus de fr.comp.os.mail, il existe d'autres forums anglais plus spécifiques. Leur nom sont explicites :

```
comp.mail.elm          Le logiciel de courrier électronique elm.
comp.mail.mh           Le logiciel Rand Message Handling.
comp.mail.mime         Multipurpose Internet Mail Extensions.
comp.mail.misc         Groupe général sur le courrier électronique.
comp.mail.multi-media  Le courrier électronique multimédia.
comp.mail.mush         Le Mail User's Shell (MUSH).
comp.mail.sendmail     L'agent sendmail de BSD.
comp.mail.smail        L'agent de courrier électronique smail.
comp.mail.uucp         Le courrier électronique dans un environnement uucp.
```

2.3 Listes de diffusion

Il existe plusieurs listes de diffusion à propos de sendmail, smail et qmail.

Vous pouvez trouver les adresses dans `/usr/doc/celui_que_vous_avez_choisi`.

2.4 Autres documents du Projet de documentation Linux (LDP)

De nombreuses autres excellentes informations sont disponibles dans les autres HOWTO Linux et via le Projet de documentation Linux.

En particulier, vous pourriez jeter un oeil à :

- le contenu du dossier /usr/doc de votre ordinateur :-) ;
- le *Linux Networking Administrators' Guide* ;
- le *Mail-User-HOWTO* ;
- le *Serial-HOWTO* ;
- le *Ethernet-HOWTO* ;
- le *UUCP HOWTO* si vous utilisez UUCP.

2.5 Livres

Voilà une liste non exhaustive de livre qui peuvent vous aider :

- *Managing UUCP and USENET* des éditions O'Reilly and Associates est à mon avis le meilleur ouvrage disponible pour comprendre les programmes et les protocoles utilisés par les sites USENET.
- *Unix Communications* de Waite Group contient une description de chacun des éléments (et plus) et de la manière dont ils s'imbriquent.
- *Sendmail* des éditions O'Reilly and Associates semble être la référence ultime sur sendmail-v8 et sendmail+IDA. C'est un incontournable pour quiconque souhaite comprendre sendmail sans se casser les dents.
- *The Internet Complete Reference* d'Osborne est un bon livre de référence qui explique les différents services disponibles sur Internet. C'est une grande source d'information sur les forums de discussions, le courrier électronique, et beaucoup d'autres sujets.
- *The Linux Networking Administrators' Guide* d'Olaf Kirch du Linux Documentation Project est disponible sur le net et est publié par (au moins) SSC et O'Reilly (NDT : existe en version française sous le titre « Administration réseau sous Linux »). Si vous ne devez en lire qu'un, celui-ci constitue un bon choix pour tout apprendre sur les réseaux Unix.

3 Comment marche le courrier électronique

Nous allons maintenant nous intéresser au flux d'informations qui se produit classiquement quand deux personnes communiquent par courrier électronique. Supposons qu'Alice, sur sa machine *paysdesmerveilles.com* veuille envoyer un courrier électronique à Bob, sur sa machine *dobbs.com*. Les deux machines sont connectées à l'Internet.

Tout d'abord il faut savoir qu'un courrier électronique est divisé en deux parties séparées par une ligne vide : les entêtes et le corps. Les entêtes contiennent la source et la destination du courrier, le sujet, la date d'envoi et d'autres informations utiles. Le corps est le contenu du message. Voici un exemple :

```
From: "Alice" <alice@paysdesmerveilles.com>
Message-Id: <199711131704.MAA18447@paysdesmerveilles.com>
Subject: Avez-vous vu mon lapin blanc ?
To: bob@dobbs.com (Bob)
Date: Thu, 13 Nov 1997 12:04:05 -0500 (EST)
Content-Type: text

Je suis extrêmement préoccupée.  J'ai peur qu'il ait chuté dans un trou.
--
                                                >>alice>>
```

La disposition et le sens des entêtes sont définis dans un standard Internet :

RFC822 `ftp://ftp.isi.edu/in-notes/rfc822.txt` .

3.1 Courrier entre des machines connectées en permanence à Internet

Voici un diagramme du processus entier (j'en expliquerai ensuite toutes les étapes et la terminologie) :

```
                   +--------+          +--------+
+-------+ compose  |        | appelle  |        |
| Alice |--------->|  MUA   |--------->|  MTA   |:::>:::::
+-------+          |appelant|          |appelant|       ::   sur la
                   +--------+          +--------+       ::   machine
                                                        ::   appelante
.......................................................................
                             SMTP                       ::
 ::::::::::::::::::::::::::::<::::::::::::::::::::::::::::
 ::
 ::   +---------+          +-----+                +-------+
 ::   |         | appelle  |     |  distribue à   | boîte |
 ::::>|   MTA   |--------->| LDA |===============>|  de   |  sur la
      |recevant |          |     |                |  Bob  |  machine
      +---------+          +-----+                +-------+  recevante
                                                    |   |
                                                    |   |
                     +----------------<-------------+   |
                     |                                  |
              +-----------+         +--------+          |
              | signaleur |         | MUA de |<---------+
              |  de Bob   |         |  Bob   |
              +-----------+         +--------+
                     |                  |
                     |      +-----+     |
                     +----->| Bob |<----+
                            +-----+
```

Pour envoyer un courrier, Alice se sert d'un programme appelé un agent d'utilisateur de courrier (ou MUA pour *Mail User Agent*). L'agent d'utilisateur de courrier est ce que l'utilisateur appellerait le « lecteur de courrier » ; il aide à composer le message, généralement en appelant l'éditeur de votre choix. Après avoir appuyé sur le bouton *envoyer*, Alice n'interviendra plus dans le processus. Plus tard dans ce HOWTO, nous examinerons les lecteurs de courriers les plus populaires.

L'agent utilisateur de courrier utilisé par Alice confie immédiatement son message à un programme appelé agent de transport de courrier (ou *MTA* pour *mail transport agent*)/. En général, ce programme sera *sendmail*, bien que quelques autres agents de transport deviennent de plus en plus populaire et puissent faire leur apparition dans de futures distributions Linux. Plus loin dans ce HOWTO, nous examinerons également le panorama des agents de transport.

Le travail de l'agent de transport est d'envoyer le courrier à l'agent de transport de la machine de Bob. Il trouve la machine de Bob en analysant l'entête *To* et en trouvant *dobbs.com* à la droite de l'adresse de Bob. Les mécanismes par lesquels s'effectue cette connexion sont un tout autre sujet ; dans le cadre de notre explication, retenez seulement que cette connexion est une manière pour l'agent de transport d'Alice d'envoyer des commandes textes à la machine de Bob et recevoir de réponses à ces commandes.

Les commandes de l'agent de transport ne vont pas dans un shell. À la place, elles sont dirigés vers un port de service sur la machine de Bob. Un port de service est une sorte de point de rendez-vous, une endroit fixe où un serveur écoute les requêtes entrantes. Les ports de service sont numérotés, et l'agent de transport de courrier d'Alice sait qu'il doit parler sur le port 25 de la machine de Bob pour envoyer le courrier.

Sur le port 25, la machine de Bob a son propre agent de transport qui écoute les commandes (probablement un autre sendmail). L'agent de transport d'Alice dialoguera avec celui de Bob en utilisant le protocole *Simple Mail Transfer Protocol* (ou *SMTP*). Voici à quoi un dialogue SMTP ressemble. Les lignes envoyées par la machine d'Alice sont marqués par un *E:*, celle de la machine de Bob par un *R:*.

```
E: MAIL FROM:<alice@paysdesmerveilles.com>
R: 250 OK
E: RCPT TO:<bob@dobbs.com>
R: 250 OK
E: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
E: From: "Alice" <alice@paysdesmerveilles.com>
E: Message-Id: <199711131704.MAA18447@paysdesmerveilles.com>
E: Subject: Avez-vous vu mon lapin blanc ?
E: To: bob@dobbs.org (Bob)
E: Date: Thu, 13 Nov 1997 12:04:05 -0500 (EST)
E: Content-Type: text
E:
E: Je suis extrêmement préoccupée.  J'ai peur qu'il ait chuté dans un trou.
E: --
E:                                                >>alice>>
E: .
R: 250 OK
```

D'habitude, une commande SMTP est une simple ligne de texte, et il en va de même pour la réponse. La commande DATA est une exception : après avoir lu la commande DATA, le serveur SMTP accepte toutes les lignes jusqu'à qu'il voit une ligne ne contenant qu'un point. (SMTP est défini dans le *RFC821* `ftp://ftp.isi.edu/in-notes/rfc821.txt` .)

Arrivé à ce point, l'agent de transport de courrier de Bob a le message d'Alice. Il va ajouter un entête au message qui ressemble à quelque chose de ce genre :

```
Received: (from alice@paysdesmerveilles.com)
        by mail.dobbs.com (8.8.5/8.8.5) id MAA18447
        for bob@dobbs.com; Thu, 13 Nov 1997 12:04:05 -0500
```

Cela permet de suivre les messages à la trace en cas d'erreur (parfois un message est relayé par plus d'un serveur et contiendra plusieurs entêtes de ce type). L'agent de transport de Bob enverra le message à un *agent de distribution local* (ou *LDA* pour *local delivery agent*). Sur les systèmes Linux, l'agent de distribution est souvent un programme appelé *procmail*, bien qu'il en existe d'autres.

Le travail de l'agent de distribution local est d'ajouter le message à la boîte aux lettres de Bob. Il est séparé de l'agent de transport pour que les deux programmes soient plus simples et pour que l'agent de transport puisse se concentrer sur la partie Internet du processus, sans avoir à se préoccuper des détails locaux comme de savoir où se trouvent les boîtes aux lettres des utilisateurs.

La boîte aux lettres de Bob sera normalement un fichier appelé `/usr/spool/mail/bob` ou `/var/mail/bob`. Lorsque Bob lit son courrier, il lance son propre agent d'utilisateur de courrier qui regarde et édite ce fichier.

3.2 Signaleur de courrier

Un autre type de programme joue un rôle important dans la chaîne de distribution du courrier électronique, bien qu'il n'envoie pas ou ne lise pas de courrier. C'est le *signaleur de courrier*, un programme qui surveille votre boîte aux lettres et vous signale l'arrivée de chaque nouveau courrier électronique.

Le signaleur de courrier original était composé de deux programme Unix appelés biff(1) et comstat(8). Biff est une interface utilisateur de comstat qui vous permet d'activer le service comstat. Quand ce service est activé, l'entête des nouveaux courriers est redirigé vers votre terminal dès qu'ils arrivent. Cette commodité a été développée pour les utilisateurs de programmes orientés lignes sur écrans cathodiques ; ce n'est pas vraiment une bonne idée dans les environnements d'aujourd'hui.

La plupart des shells intègrent des facilitées équivalentes, mais de manière moins intrusive (en émettant un message juste avant l'invite lorsqu'un nouveau message est détecté). En général, on active cette fonction en définissant une variable d'environnement documentée dans la page de manuel du shell. Pour les shells de la famille sh/ksh/bash, regardez les variables MAIL et MAILPATH.

Les systèmes utilisant X sont livrés avec l'un des nombreux petits gadgets qui vérifient périodiquement si de nouveaux courriers sont arrivés, et vous le signale par des messages sonores et visuels. Le plus vieux et le plus utilisé est appelé *xbiff* ; si notre système Linux a un bureau X préconfiguré, xbiff est probablement dessus. Reportez-vous à la page de manuel xbiff(1) pour plus de détails.

3.3 Courrier vers des machines connectées occasionnellement à Internet

Si vous avez lu attentivement, vous avez peut-être remarqué que les échanges d'informations que nous avons décrit ci-dessus dépendent du fait que la machine d'Alice peut immédiatement parler à la machine de Bob. Que se passe-t-il si la machine de Bob est en panne, ou si elle n'est pas connectée à Internet ?

Si l'agent de transport d'Alice ne peut pas atteindre celui de Bob immédiatement, il va stocker le message d'Alice dans une file d'attente sur *paysdesmerveilles.com*. Il essaiera ensuite de renvoyer le message à des intervalles de temps réguliers, jusqu'à ce qu'un temps d'expiration soit dépassé. À ce moment, il renverra son message à Alice en l'avertissant du problème. Dans la configuration par défaut de l'agent de transport le plus populaire (sendmail), l'intervalle entre chaque essai de transmission est de 15 minutes et la durée d'expiration de 4 jours.

3.4 Courrier distant et protocole de courrier distant

Beaucoup d'utilisateurs de Linux sont connectés à Internet via un fournisseur d'accès Internet (FAI) et ne possèdent pas leur propre domaine. À la place, ils ont un compte sur une machine de leur FAI. Leurs courriers électroniques sont distribués dans une boîte aux lettres sur cette même machine. Cependant, généralement ces utilisateurs veulent lire et répondre à leurs courriers en utilisant leur propre machine qui est connecté à leur FAI de manière intermittente en utilisant *SLIP* ou *PPP*. Linux permet d'utiliser des protocoles de courrier distant qui permettent cela.

Remarquez comme ceci est différent du scénario que nous avons discuté dans la première partie. Le courrier était placé dans une file en attendant d'être envoyé, alors que dans le cas que nous venons de voir, le courrier est envoyé directement dans une boîte aux lettres sur un serveur. Le courrier dans une file n'est pas considéré comme ayant été distribué et peut expirer tandis que le courrier distribué dans la boîte aux lettres du serveur de courrier d'un FAI est considéré comme distribué et peut rester bloqué sur ce serveur indéfiniment.

Un protocole de courrier distant permet au courrier sur un serveur d'être récupéré au travers d'un lien réseau par un programme client (c'est le contraire d'une distribution normale au cours de laquelle un agent de transport envoie lui-même le courrier vers un autre agent de transport). Il existe deux protocoles communément utilisés de courrier à distance définis par les standards Internet : POP3 (*RFC1939* `ftp://ftp.isi.edu/in-notes/rfc1939.txt`) et IMAP (*RFC2060* `ftp://ftp.isi.edu/in-notes/rfc2060.txt`). Tous les FAI permettent d'utiliser POP3 et un nombre croissant permet également IMAP (qui est plus puissant).

Voilà à quoi ressemble une session POP3 :

```
S: <le client se connecte au port de service 110>
R:    +OK POP3 server ready <1896.697170952@mailgate.dobbs.org>
S:    USER bob
R:    +OK bob
S:    PASS redqueen
R:    +OK bob's maildrop has 2 messages (320 octets)
S:    STAT
R:    +OK 2 320
S:    LIST
R:    +OK 2 messages (320 octets)
R:    1 120
R:    2 200
R:    .
S:    RETR 1
R:    +OK 120 octets
R:    <le serveur POP3 envoie le message 1>
R:    .
S:    DELE 1
R:    +OK message 1 deleted
S:    RETR 2
R:    +OK 200 octets
R:    <le serveur POP3 envoie le message 2>
R:    .
S:    DELE 2
R:    +OK message 2 deleted
S:    QUIT
R:    +OK dewey POP3 server signing off (maildrop empty)
S:  <le client se déconnecte>
```

Une session IMAP utilise des commandes et des réponses différentes, mais est très similaires.

Pour tirer parti de POP3 ou IMAP vous avez besoin d'un client de courrier à distance pour retirer votre courrier. Certains lecteurs de courrier offrent nativement ce genre de fonctionnalités. C'est le cas de Netscape Messenger qui permet d'utiliser POP et IMAP.

Le principal inconvénient des lecteurs de courrier implémentant ce genre de fonctionnalités est que vous devez explicitement dire à votre lecteur de retirer votre courrier sur le serveur ; vous ne pourrez pas, par exemple, être averti par xbiff(1) comme vous le seriez si le courrier était local ou distribué par un serveur SMTP. De plus, bien sûr, tous les MUA n'implémentent pas POP ou IMAP, et vous devrez vous rabattre sur d'autres fonctionnalités.

Votre distribution contient probablement un programme appelé *fetchmail* . Il a été écrit spécialement pour dialoguer avec des serveurs de courrier distant, récupérer le courrier, et le distribuer dans votre répertoire de réception de courrier, en dialoguant avec votre serveur SMTP.

Mis à part si vous avez besoin de laisser vos courriers sur le serveur (par exemple parce que vous changez souvent de machine) fetchmail est probablement une solution meilleure que n'importe quelle lecteur de courrier intégrant POP ou IMAP. Fetchmail peut se lancer en arrière plan et récupérer votre courrier périodiquement, et votre xbiff(1) ou tout autre signaleur de courrier marchera comme si tout se passait par SMTP. De plus, fetchmail est beaucoup plus tolérant que les lecteurs de courrier envers les différents particularismes et non respect des standards des serveurs de messagerie, et il gère mieux les erreurs.

Voici un diagramme schématisant les deux cas (avec et sans fetchmail) :

```
                   +--------+          +--------+
+-------+ compose  |        | appelle  |        |
| Alice |--------->|  MUA   |--------->|  MTA   |:::>:::::
+-------+          |appelant|          |appelant|       ::   sur la
                   +--------+          +--------+       ::   machine
                                                        ::   appelante
                          SMTP                          ::
 ::::::::::::::::::::::::::::<::::::::::::::::::::::::::::
 ::
.:........................................................................
 ::
 ::   +---------+          +-----+             +---------+
 ::   |         | appelle  |     | distribue à |boîte du |
 ::::>|   MTA   |--------->| LDA |============>| serveur |::::>::::
      |recevant |          |     |             | de Bob  |       ::  sur le
      +---------+          +-----+             +---------+       ::  serveur de
                                                                 ::  courrier
                          POP ou IMAP                            ::
 ::::::::::::::::::::::::::::<:::::::::::::::::::::::::::::::::::::
 ::
.:.........................................................................
 ::
 ::                  +-----------+
 ::                  |           |
 ::::::>:::::::::::::| fetchmail |::::::::               sur la
 ::                  |           |      ::               machine
 ::                  +-----------+      ::               recevante,
 ::                                     ::               avec fetchmail
 ::   ::::::::::::::::<:::::::::::::::::::
```

```
 ::   ::
 ::   ::   +---------+         +-----+                +-------+
 ::   ::   |         | appelle  |     |  distribue à   | Boîte |
 ::   ::::>|   MTA   |--------->| LDA |===============>|  de   |
 ::        |recevant |          |     |                |  Bob  |
 ::        +---------+          +-----+                +-------+
 ::                                                      |   |
 ::                                                      |   |
 ::                       +----------------<-------------+   |
 ::                       |                                  |
 ::                  +-----------+         +--------+        |
 ::                  | signaleur |         | MUA de |<--------+
 ::                  |  de Bob   |         |   Bob  |
 ::                  +-----------+         +--------+
 ::                       |                  |
.::..........................................................................
 ::                   .   |                  |
 ::     sans          .   |                  |
 ::    fetchmail      .   |                  |
 ::                   .   |      +-----+     |
 ::   +----------+    .   +----->|     |<----+
 ::   |    MUA   |    .          | Bob |
 :::::| POP/IMAP |----.--------->|     |
      |  de Bob  |    .          +-----+
      +----------+    .
```

3.5 Format des boîtes aux lettres

Quand le courrier entrant est ajouté à une boîte aux lettres, il est du ressort de l'agent de transport de courrier d'ajouter des sortes de délimiteurs qui indiquent où un courrier s'arrête et où le suivant débute.

Sous Unix, que pratiquement tous les lecteurs de courrier suivent la même convention. Chaque ligne commençant par *From* (attention, il y a un espace après *From*) débute un nouveau courrier. Si un *From* apparaît dans une ligne de texte, un agent de transport de courrier d'Unix ajoutera généralement un signe supérieur-à au début de la ligne (>*From*). Après cette ligne (qui continue généralement par le nom de l'envoyeur et la date de réception) on trouve l'entête RFC822 du courrier.

Cette convention date d'Unix Version 7, et donc on parle pour ce type de boîtes aux lettres de boîtes aux lettres V7 (*V7 mailboxes*). Sauf quand ce sera indiqué, tous les programmes mentionnés dans ce HOWTO utilisent ce format. Il n'est cependant pas universel et des outils attendant et générant des formats différents peuvent se placer mutuellement dans une situation de confusion catastrophique.

Les quatre autres formats à connaître (il faut s'en méfier !) sont BABYL, MMDF, MH et qmail maildir. De cela, MMDF est le plus simple : il utilise des délimiteurs (quatre fois le caractère ASCII 001 suivi par CR-LF). MMDF fût un des premiers formats de boîtes à lettres et il est plutôt cru. Un dérivé est encore utilisé sur les systèmes SCO.

BABYL est un autre survivant d'un vieux système de courrier du MIT. Il est encore utilisé par le mode lecteur de courrier d'Emacs.

Les formats MH et qmail maildir utilisent, pour stocker chaque message, un fichier séparé. Les boîtes aux lettres sont représentées par des répertoires. Faire une recherche dans une telle boîte aux lettres demandera l'utilisation d'un *grep -r* (grep récursif) afin d'accéder aux messages contenus dans la boîte aux lettres.

Les boîtes aux lettres .mbx de Microsoft Outlook Express peuvent être converties au format RFC822 grâce à *mbx2mbox*.

4 Configuration requise

4.1 Matériel

La mise en oeuvre du courrier électronique sous Linux ne nécessite pas de disposer d'un matériel particulier.

Vous aurez besoin d'un logiciel de « transport » quelconque pour vous connecter à des systèmes distants, c'est-à-dire soit TCP/IP, soit UUCP.

Cela signifie que vous aurez besoin d'un modem ou d'une carte Ethernet selon votre configuration.

Dans la plupart de cas, vous préférerez avoir le modem le plus rapide que vous puissiez utiliser, à savoir un V90 à 57 600 bit par secondes au moment où j'écris ce HOWTO. En général, vous préférerez avoir un UART 16550 sur votre carte série ou dans votre modem pour gérer les vitesses supérieurs à 9600 bauds.

Si vous ne comprenez pas cette dernière phrase, consultez le groupe *comp.dcom.modems* ou les diverses excellentes FAQ sur les modems et la communication série ainsi que les messages périodiques sur USENET.

5 Choisir un agent de transport de courrier (MTA)

L'agent de transport de courrier est le programme qui transfère le courrier de votre système local aux systèmes distants. Il est très rarement nécessaire d'avoir à trifouiller ou remplacer votre agent de transport sur les derniers systèmes Linux, et vous feriez mieux de ne pas chercher à réparer ce qui n'est pas défectueux. Néanmoins, voici un exposé pour vous permettre de comprendre les changements à faire si vous décidez que vous avez besoin d'une plus grande sécurité ou de meilleures performances que votre système ne peut vous en offrir par défaut.

(Il y a d'autres agents de transport Unix que ceux présentés ci-dessous, mais vous n'avez pratiquement aucune chance de les rencontrer sur une machine sur laquelle tourne Linux.)

Chacun de ces agents a ses propres fonctionnalités, mais le meilleur compromis est qmail. Il est très sécurisé (même si vmail l'est encore plus), très rapide (même si smail est plus rapide pour les utilisateurs locaux) et facile à configurer. Bien sûr, sentez-vous libre de choisir n'importe lequel d'entre eux. Les informations fournies ici ont pour but de vous aider à bien choisir.

Sendmail peut être bien pour de nombreux sites ayant besoin d'options compliquées mais je crois que sa configuration est trop difficile pour un débutant alors qu'il n'est ni très sécurisé ni très rapide. C'est pourquoi il y a seulement une section *vraiment* dépassée sur sendmail dans ce HOWTO.

Si vous savez ce que vous faîtes, choisissez sendmail (et vous ne devriez pas être en train de lire ce HOWTO !) ; sinon, je recommande généralement qmail.

Voici une description détaillée de ces programmes.

5.1 sendmail

BSD sendmail est le grand-père des agent de transport d'Internet. Il a enterré pas mal de ceux qui auraient dû être ses successeurs. La plupart des distributions Linux l'utilise maintenant et le préinstalle.

sendmail a depuis très longtemps la réputation d'être un cauchemar pour les administrateurs — difficile à comprendre, compliqué à configurer et plein de trous de sécurité. Cependant, avec la stabilisation des technologies et des standards d'Internet, beaucoup des options et des règles configurables de sendmail qui sont à l'origine de cette réputation ont cessé de requérir des bidouillages propres à chaque site (l'abandon des couches réseaux non TCP/IP comme UUCP y est pour beaucoup). De plus, les versions récentes de sendmail ont un système de configuration amélioré qui permet de s'épargner la légendaire laideur du fichier de configuration `sendmail.cf`. Plus important, sendmail est maintenant préconfiguré de sorte que nous n'ayez à y toucher que si vous avez une configuration peu commune (comme par exemple le routage du courrier sur un réseau non TCP/IP).

Il y a un *site sendmail* `http://www.sendmail.org` . Il comporte des références à la documentation étendue de sendmail avec laquelle vous devrez vous battre pour peaufiner votre configuration.

D'autres agents de transport peuvent prendre le nom de sendmail et imiter la sémantique des options passées sur la ligne de commande de sendmail. C'est pratique pour les lecteurs de courrier qui supposent souvent qu'ils parlent avec sendmail.

5.2 smail v3.2

smail fut la première sérieuse tentative de remplacer sendmail. Il a un système de configuration plus simple et beaucoup plus compréhensible que celui de sendmail et est plutôt bien sécurisé. Certaines distributions Linux le préinstallent plutôt que sendmail.

À un moment, l'excellent support pour les sites utilisant TCP/IP et UUCP était un argument majeur pour lui, mais dés lors que UUCP fut de moins en moins utilisé, il en fut de même pour smail. De plus, smail est moins efficace que sendmail sur les gros volumes de connexion.

Comme avec sendmail, il est peu probable que vous ayez besoin de trifouiller une configuration de smail préinstallée.

(Très occasionnellement il se peut que vous trouviez des références à *smail 2.5*. Ce programme est obsolète depuis longtemps. Ne vous embêtez pas avec lui.)

5.3 qmail

Le programme *qmail* `http://pobox.com/~djb/qmail.html`

est un agent de transport de courrier compatible avec sendmail écrit spécialement dans le but d'être très sécurisé. L'auteur a promis une récompense de 500 dollars pour celui qui trouverait le premier véritable trou de sécurité ; cette récompense n'a toujours pas été demandé depuis mars 1997.

5.4 exim

Le programme *exim* `http://www.exim.org/` est similaire à smail3 mais avec plus de fonctionnalités. Il est particulièrement efficace pour l'élimination du spam et supporte plusieurs adresses virtuelles (*virtual DNS domains*) sur la même adresse.

Je l'ai essayé sur mon ordinateur, il semble être une fusion du système de configuration de smail et de la sécurité de qmail. De plus il a l'avantage d'être sous GPL.

Une section expliquant comment remplacer votre agent de transport de courrier par exim sera bientôt ajoutée.

6 Installation de l'agent de transport de courrier (MTA)

6.1 Qmail v1.03

Sécurisé, rapide et facile à utiliser, c'est mon agent de transport de courrier préféré.

Actuellement, aucune distribution ne préinstalle qmail. Nous allons nous concentrer sur la compilation et l'installation de qmail puisque c'est la seule partie difficile : la configuration est vraiment immédiate.

6.1.1 Obtenir qmail

Allez sur `www.qmail.org` pour télécharger la dernière version.

6.1.2 Décompresser les sources

Ensuite, décompressez le en tapant

```
mv qmail.tar.gz /usr/local/src
cd /usr/local/src ; tar -zxvf qmail.tar.gz
```

Si vous trouvez une version bz2 (qui est un nouveau et meilleur format de compression), remplacer tar par

```
bunzip2 qmail.tar.bz2
tar -xvf qmail.tar
```

6.1.3 Préparer la compilation

Maintenant entrez dans le répertoire qmail pour examiner la configuration par défaut :

```
cd qmail; more conf-*
```

Vous ne devriez avoir à changer aucun paramètre, mais vous pouvez (par exemple) spécifier un répertoire d'installation alternatif ou de meilleurs options de compilation.

Maintenant lancer

```
mkdir /var/qmail
```

pour créer le répertoire d'installation.

Si nous n'avez pas installé une distribution Debian, vous devrez ajouter plusieurs UID pour l'utilisation de qmail. La grande sécurité de qmail dépend de cela.

Le fait que qmail soit divisé en modules tournant chacun sous leur propre UID rend la tâche de mettre hors-service la totalité de votre système de courrier ou de gagner un accès root plus difficile.

Donc, lancer

```
# groupadd nofiles
# useradd -g nofiles -d /var/qmail/alias alias
# useradd -g nofiles -d /var/qmail qmaild
# useradd -g nofiles -d /var/qmail qmaill
# useradd -g nofiles -d /var/qmail qmailp
# groupadd qmail
```

```
# useradd -g qmail -d /var/qmail qmailq
# useradd -g qmail -d /var/qmail qmailr
# useradd -g qmail -d /var/qmail qmails
```

ou éditer à la main `/etc/passwd` et `/etc/group` pour ajouter ces utilisateurs vous-même.

Evan E. m'a signalé qu'il avait dû utiliser le paramètre *-g GID* avec la version officielle de groupadd (Caldera 1.2) car sinon groupadd retournait cette erreur : « A group with that name already exists. » (un groupe de ce nom existe déjà).

Par exemple, vous pouvez respectivement ajouter

```
qmail:*:2107:
nofiles:*:2108:
```

et

```
alias:*:7790:2108::/var/qmail/alias:/bin/true
qmaild:*:7791:2108::/var/qmail:/bin/true
qmaill:*:7792:2108::/var/qmail:/bin/true
qmailp:*:7793:2108::/var/qmail:/bin/true
qmailq:*:7794:2107::/var/qmail:/bin/true
qmailr:*:7795:2107::/var/qmail:/bin/true
qmails:*:7796:2107::/var/qmail:/bin/true
```

Maintenant vous pouvez lancer

```
make setup check
```

pour tester votre configuration, ensuite

```
./config
```

pour configurer qmail.

Attention, votre serveur disposer d'une adresse reconnue (i.e. pouvant être résolue par une requête DNS) ou `./config` sera perturbé.

Si ce n'est pas le cas, vous pouvez donnez le nom de votre serveur directement via :

```
./config-fast ordinateur.lambda.fr
```

Maintenant, vous devez installer quelques alias, puisque `/etc/alias` n'est pas utilisé pas qmail sauf si vous compilez et installez un paquetage optionnel.

Voici ma configuration :

```
File : ".qmail-MAILER-DAEMON"
&postmaster
File : ".qmail-bin"
&root
File : ".qmail-daemon"
&root
File : ".qmail-decode"
&root
```

```
File : ".qmail-dumper"
&root
File : ".qmail-games"
&root
File : ".qmail-ingres"
&root
File : ".qmail-mailer-daemon"
&postmaster
File : ".qmail-manager"
&root
File : ".qmail-news"
&root
File : ".qmail-nobody"
&root
File : ".qmail-operator"
&root
File : ".qmail-postmaster"
&root
File : ".qmail-root"
&guylhem
File : ".qmail-system"
&root
File : ".qmail-toor"
&root
File : ".qmail-uucp"
&root
File : ".qmail-uucp-default"
|preline -dr /usr/bin/uux - -r -gC -a"${SENDER:-MAILER-DAEMON}" lm!rmail "($DEFAULT@$HOST)"
```

Vous devez créer chacun de ces fichiers dans `~alias` en remplaçant *&guylhem* dans `.qmail-root` par votre propre compte récupérant le courrier de root.

Attention si vous utilisez UUCP !

Ne faites pas confiance à la FAQ de qmail pour UUCP, utilisez mon `.qmail-uucp-default` à la place sinon vous ne pourrez envoyer aucun courrier par votre connexion UUCP !

Maintenant vous devez décider dans quel format vos utilisateurs recevront leur courrier.

Voici mon avis :

- Pour des répertoires *home* montés en NFS, utilisez le format MAILDIR avec un correctif pour les logiciel de courrier (les correctifs sont disponibles sur `www.qmail.org`).
- Si aucun correctif n'est disponible, préférez le format MAILFILE. N'importe quel logiciel peut lire un fichier à ce format. Les utilisateurs devront seulement créer un alias (pour bash) ou un setenv (pour csh) pour leur logiciel de courrier.
- Éviter le format `/var/spool/mail/$USER` format, trop peu sûr.

Pour corriger le format par défaut, lisez chacun des fichiers dans `/var/qmail/boot` et copiez celui que vous préférez vers `/var/qmail/rc`.

`home` ou `proc` sont des choix sûrs mais je préfère `home` pour des raisons de sécurités.

6.1.4 Configurer qmail

Dans `/var/qmail/control`, éditez :

defaultdomain, me, plusdomain

- *me* est votre nom de domaine qualifié complet (*full qualified domain name*) local, par exemple sur ma machine c'est *barberouge.linux.lmm.com*
- *defaultdomain* sera ajouté à toutes adresses sans point, y compris l'adresse par défaut (*me*). Par exemple vous pouvez le positionner à *reseau-local* et les mails envoyés à *pierre@sa-boite* seront complétés et envoyés à *pierre@sa-boite.reseau-local*.
- *plusdomain* est l'exception : il est ajouté à toutes les adresses qui se termine par un signe plus, y compris l'adresse par défaut.

Ces trois exemples vous montre la puissance et la facilité de la configuration de qmail !

locals, rcpthosts Si vous voulez utiliser les noms de domaines virtuels, ajoutez simplement les noms des domaines dans ces fichiers. Tout courrier reçu envoyé à ces noms sera traité localement.

La différence entre *locals* et *rcpthosts* est que *rcphosts* n'est pas considéré comme un alias local, ce qui est utile si vous recevez des courriers électroniques envoyés à une adresse gratuite comme yahoo.com ou lemel.fr alors que vous envoyez également des courriers à des utilisateurs de ces services non locaux et que vous ne voulez pas traiter localement un courrier envoyé à quelqu-un@yahoo.com !

virtualdomains Ici vous pouvez spécifier le mode d'envoi par défaut, par exemple

```
#:alias-uucp
```

si vous ne voulez pas envoyer vos mail par UUCP mais par SMTP (c'est la configuration par défaut) ou

```
:alias-uucp
```

si vous voulez les envoyer par UUCP.

6.1.5 Tester qmail

Maintenant, qmail est configuré, essayez

```
sh -cf '/var/qmail/rc &'
```

pour lancer qmail (ça n'interférera pas avec votre agent de transport de courrier), ensuite

```
echo to: mon-identifiant | /var/qmail/bin/qmail-inject
```

Vous devriez recevoir ce mail dans le format que vous choisi dans */var/qmail/boot/*.

6.1.6 Désinstaller votre ancien agent de transport de courrier

Si le test est concluant, tuez votre ancien agent de transport.

Tout d'abord, arrêtez-le :

```
$ killall -STOP nom_du_démon
```

Si des processus-fils tournent, forcez-les à se terminer :

```
$ killall -CONT leur_nom
```

Puis réessayez de stopper votre agent de transport de courrier (s'il le faut, répétez ces deux étapes *ad nauseam*).

Ensuite, tuez votre agent de transport :

```
$ killall -TERM nom_du_démon
$ killall -CONT nom_du_démon
```

Virez-le. La manière de le faire dépend de votre distribution. Par exemple `rpm -e -nodeps` pour une RedHat, une Caldera ou une SuSE, ou `dpkg -r -force-depends` pour une Debian. Lancer ensuite

```
# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail
# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

Maintenant configurer qmail-smtpd dans `/etc/inetd.conf` :

```
smtp stream tcp nowait qmaild /var/qmail/bin/tcp-env tcp-env /var/qmail/bin/qmail-smtpd
```

Tout ceci doit se trouver sur la même ligne.

Si vous utilisez un vielle distribution avec un init non SysV (c'est par exemple le cas des vielles RedHat), ajoutez cela à vos scripts d'initialisation :

```
sh -cf '/var/qmail/rc &'
```

Il y a de grandes chances pour que ce soit `/etc/rc.local` mais cela peut varier.

Pour les distribution actuelles avec des init SysV (RedHat, Caldera, SuSE, Debian), ajoutez ce script dans `/etc/init.d/` ou `/etc/rc.d/init.d` :

- Pour la Debian :

```
#!/bin/sh

test -x /var/qmail/rc || exit 0

case "$1" in
  start)
     echo -n "Starting mta: " # démarrage de l'agent de transport
     sh -cf '/var/qmail/rc &'
     echo "qmail."
     ;;
  stop)
     echo -n "Stopping mta: " # arrêt de l'agent de transport
     killall qmail-lspawn
     echo "qmail."
     ;;
  restart)
     echo -n "Restarting mta: " # redémarrage de l'agent de transport
     killall -HUP qmail-lspawn
```

```
        killall -ALRM qmail-lspawn
        echo "qmail."
        ;;
    *)
        echo "Usage: /etc/init.d/qmail {start|stop|restart}"
        exit 1
esac

exit 0
```

- Pour la RedHat :

```
#!/bin/sh
#
# qmail         ce script shell s'occupe du démarrage et de l'arrêt de qmail
#
# description : qmail est un agent de transport de courrier, c'est-à-dire
#               le programme qui s'occupe de transporter le courrier d'une
#               machine à une autre
# nom de processus : qmail
# config: /var/qmail/control/

# source la bibliothèque de function
. /etc/rc.d/init.d/functions

# source la configuration du réseau
. /etc/sysconfig/network

export PATH=$PATH:/var/qmail/bin

# vérifie que le réseau est en place
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/sbin/sendmail ] || exit 0

# examine les options de démarrage du script
case "$1" in
  start)
        # Lance les démons.
        echo -n "Starting qmail: " # démarrarage de qmail
        qmail-start '|preline procmail' splogger qmail &
        touch /var/lock/subsys/qmail
        echo
        ;;
  stop)
        # Arrête les démons.
        echo -n "Shutting down qmail: " # arrêt de qmail
        killproc qmail-lspawn
        echo
        rm -f /var/lock/subsys/qmail
        ;;
  restart)
        $0 stop
        $0 start
        ;;
  status)
```

```
                status qmail
                ;;
        *)
                echo "Usage: qmail {start|stop|restart|status}"
                exit 1
esac

exit 0
```

Faîtes des liens symboliques vers chaque /etc/rc.d/rcN.d/, par exemple :

```
ln -sf /etc/init.d/qmail /etc/rc.d/rc1.d/K19qmail
```

Si la première lettre est K, vous tuerez qmail quand la machine passera à ce niveau d'exécution (1 pour le mode simple ou 6 pour le redémarrage) et si la première lettre est S, vous le démarrerez (2, 3, 4, 5).

- Comment décider si vous devez mettre un K ou un S ? Faites ce que le majorité des démons font dans ces runlevels !
- Quel nombre devrais-je mettre après K ou S ? Le nombre suivant de celui de votre démon réseau.

  Cela signifie que l'agent de transport de courrier sera démarré et arrêté respectivement avant et après votre démon réseau.

  Sans cela, votre réseau ne sera pas activé au moment où sera lancé votre agent, alors que celui-ci s'attendra à ce qu'il fonctionne.

RedHat, Caldera et SuSE utilisent `/etc/rc.d` à la place de `/etc` pour un Debian. Ainsi, sur ces distributions, quand je parle de `/etc/rc1.d` comprenez `/etc/rc.d/rc1.d`.

6.1.7 Et puis c'est tout !

Pas besoin de redémarrez (Hé, vous utilisez Linux, pas un de ces petits OS de rigolos !) pour que les modifications prennent effet. Lancez juste

```
killall inetd
init 1
```

pour passez en mode mono-utilisateur. Puis tapez

```
init 2
```

pour retourner à votre niveau d'exécution par défaut (indiqué dans /etc/inittab avec le label initdefault).

Vous auriez également pu lancer à la main le script qmail mais la « méthode init » vous permet de constater si le script qmail est bien placé, c'est-à-dire lancé après les scripts réseaux mais avant tout programme qui utilise le courrier pour vous avertir (comme inn par exemple).

6.2 Smail v3.1

Smail 3.1 est *de facto* un agent de transport de courrier standard pour les sites utilisant uniquement UUCP et pour quelques un de ceux qui utilisent SMTP. Il est facile à configurer, il se compile sans correctif et est assez bien sécurisé.

6.2.1 Configurer smail

Installez le binaire smail de votre distribution (je vous recommande cette méthode) ou récupérez les sources de smail et compilez le. Si vous compilez smail, vous aurez besoin d'avoir ce qui suit dans votre fichier `os/linux` pour que sed fabrique des scripts shell qui marchent chez vous.

```
CASE_NO_NEWLINES=true
```

Une fois installées, les fichiers de configurations iront certainement dans le répertoire `/etc/smail` (mais cela peut-être différent si vous utilisez une vielle distribution) ; commençons par les éditer !

le fichier `config`

```
# source
smart_path=polux
smart_transport=uux

# destination
hostname=barberouge
domains=linux.lmm.com

visible_name=barberouge.linux.lmm.com
uucp_name=barberouge.linux.lmm.com

# max_message_size=512k
# auth_domains=foo.bar
# more_hostnames=barberouge.polux.freenix.fr
```

Bon, premièrement, qui va vous alimenter ? Je suis alimenté par *polux* via UUCP (i.e., par transport UUX) ; naturellement vous devez changer cela en fonction de votre propre situation. Par exemple, vous pourriez être alimenté par *bargw.bar.foobar.com* via SMTP, dans ce cas vous n'avez pas besoin de fichier de transport et vous pouvez définir *-transport_file* pour indiquer que vous n'en avez pas besoin.

Vous pouvez aussi utiliser *postmaster_address = votre_nom*, cachez la topologie du réseau dans les adresses sortantes (si vous êtes un portail) en utilisant *bvisible_name*, choisir quelles adresses alias peuvent aussi être utilisées pour les courriers que vous recevez grâce à *more_hostnames*.

Voyez la documentation de smail pour plus de détails ou les exemples de `/usr/doc/smail/examples` pour voir s'il y en a qui correspondent à votre cas.

le fichier `directors`

```
# aliasinclude - développe les adresses «~:include:filename~» produites
# par les fichiers alias.
# Cette entrée et la suivante sont à peu près des modèles standards.
# Il existe peu de raisons d'y apporter des changements significatifs.
# Leur unique raison d'être est de reconnaitre et de développer les
# adresses de la forme :
#       :include:pathname
# qui peuvent apparaitre dans les fichiers alias, les listes de
# diffusions ou les fichiers forward (produite par toute
# directive dont le pilote (driver) est forwardfile).
aliasinclude:
```

```
        driver = aliasinclude,          # utiliser le pilote destiné à ce cas
                                        #   particulier ;
        nobody;                         # lorsque des violations modérées des droits
                                        #   se produisent, associer aux adresses
                                        #   l'identifiant nobody ;
        copysecure,                     # utiliser les droits du director alias ;
        copyowners,                     # utiliser le propriétaire du director alias.


# forwardinclude - développe les adresses « :include:filename » produites
# par les fichiers forward
forwardinclude:
       driver = forwardinclude,         # utiliser le pilote destiné à ce cas
                                        #   particulier ;
       nobody;
       copysecure,                      # utiliser les droits du director forwarding ;
       copyowners,                      # utiliser le propriétaire du director forwarding.

# aliases - cherche les développements d'alias stockés dans une base de donnée.
# C'est le fichier d'alias standard. Il est utilisé pour réaliser des opérations
# courantes, comme faire correspondre root, postmaster, MAILER-DAEMON et uucp aux
# administrateurs du site, créer des développements de petite envergure
# d'alias système, ou d'autres choses de ce genre. Dans la configuration de
# ce site, le fichier alias est utilisé principalement pour des informations
# d'alias et de renvois spécifiques à chaque machine.
# Les informations générales de renvoi de courrier doivent être mises dans la
# base de donnée « forward ».
aliases:
        driver=aliasfile,               # directeur d'alias généraliste ;
        -nobody,                        # toutes les adresses sont associés
                                        #   par défaut à nobody, donc définir
                                        #   cette option n'est pas utile ;
        sender_okay,                    # ne pas ignorer l'expéditeur lors des
                                        #   développements d'adresses ;
        owner=owner-$user;              # les problèmes sont dirigés vers
                                        #   l'adresse du propriétaire ;
        file=/etc/aliases,
        modemask=002,                   # l'accès en écriture ne doit pas être
                                        #   autorisé à tous ;
        optional,                       # continuer normalement si le fichier
                                        #   n'existe pas ;
        proto=lsearch,                  # fichier ASCII non trié.


# forward - cherche des développements d'adresse selon le contenu de la base
# de données « forwarding ».
# C'est la base des adresses des renvois utilisateurs pour tout un sous-domaine.
# Les informations nécessaires pour faire suivre le courrier des utilisateurs,
# présents ou passés, vers leur machine préférée sont conservées ici.
# La base forward est expédiée vers les autres systèmes locaux lors de chaque
# modification, afin de maintenir la cohérence du réseau.
#forward:
#       driver = aliasfile,             # Directeur d'alias généraliste ;
#       -nobody,                        # toutes les adresses sont associés
                                        #   par défaut à nobody, donc définir
```

```
                                        #   cette option n'est pas utile ;
#       owner = real-$user;             # les problèmes sont dirigés vers
                                        #   l'adresse du propriétaire ;
#
#       file = /etc/forward,
#       modemask = 002,
#       proto = dbm,                    # Utiliser la librairie dbm(3X)
                                        #   pour accéder à la base.


# dotforward - développe les fichiers .forward des répertoires personnels.
# Pour les utilisateurs qui ont une entrée dans la base « forward », un fichier
# .forward n'est utilisé que s'il est sur leur machine personnelle, telle
# qu'indiquée dans la base forward. Lorsque ce fichier est utilisé, il est traité
# comme une liste d'adresse indiquants la liste des destinataires auxquels

# doivent être envoyés les courriers reçus par cet utilisateur, au lieu (ou
# en plus) d'être stockés en local.
dotforward:
        driver = forwardfile,           # le directeur généraliste forwarding ;
        owner = postmaster, nobody, sender_okay;

        file = ~/.forward,              # le fichier .forward du répertoire personnel ;
        checkowner,                     # l'utilisateur peut être propriétaire de ce fichier ,
        owners = root,                  # ou bien root ;
        modemask = 002,                 # ne doit pas être accessible à tous en écriture ;
        caution = daemon:root,          # ne rien exécuter en tant que root ou daemon ;
        # faire extrêmement attention aux dossiers personnels potentiellement
        # accessibles à distance.
        unsecure = "~uucp:/tmp:/usr/tmp:/var/tmp"


# forwardto - développe le « Forward to » dans les fichiers boîtes aux lettres utilisateur.
# Cela émule le méchanisme de renvoi de courrier des systèmes V6/V7/System-V qui utilise
# une ligne d'adresses de renvoi au début des fichiers boites aux lettres (BAL) utilisateur
# avec comme préfixe la chaîne « Forward to  » (avec un blanc après to)
forwardto:
        driver = forwardfile,
        owner = postmaster, nobody, sender_okay;

        file = /var/spool/mail/${lc:user},      # pointe sur la BAL utilisateur ;
        forwardto,                      # autoriser la fonction « Forward to » ;
        checkowner,                     # l'utilisateur peut être propriétaire de ce fichier,
        owners = root,                  # ou root ;
        modemask = 0002,                # sous System V, le groupe mail a la permission d'écriture ;
        caution = daemon:root           # ne rien exécuter sous root ou daemon.


# user - associe un utilisateur sur l'hôte local à la distribution dans sa BAL.
user:   driver = user;                  # reconnait les utilisateurs locaux ;
        transport = local               # le transport local distribue le courrier
                                        #   dans les BAL.


# real_user - reconnaît les utilisateurs préfixés par la chaîne "real-"
```

```
# Très utile pour permettre à une adresse de correspondre explicitement à
# une BAL utilisateur. Par exemple, des erreurs de développement d'un fichier
#.forward pourraient être distribuées ici, ou des boucles dans le renvoi de courrier entre
# plusieurs machines pourraient être résolues en utilisant une adresse real-nom_d_utilisateur.
# Aussi, les utilisateurs qui souhaitent utiliser le courrier comme moyen de transfert de
# données vers une machine qui n'est pas leur machine personnelle peuvent envoyer un
# courrier à real-nom_d_utilisateur@hôte-distant.
real_user:
        driver = user;
        transport = local,
        prefix = "real-"                # par exemple, reconnait real-root.

# lists - développe les listes de diffusion stockées dans le répertoire list
# Les listes de diffusions peuvent être crées simplement en créant un fichier
# dans le répertoire /etc/smail/lists.
lists:  driver = forwardfile,
        caution,                        # marque toutes les adresses avec l'indicateur « caution »,
        nobody,                         # et les associent à l'utilisateur nobody ;
        owner = owner-$user;            # les sites system V peuvent vouloir utiliser
                                        #   o-$user, puisque owner-$user peut être
                                        #   trop long, la taille limite des noms de
                                        #   fichiers étant de 14 caractères ;
        file = lists/${lc:user}         # les listes sont sous $smail_lib_dir.

# owners - développe les listes de diffusion stockées dans le répertoire des
#          propriétaires de listes (répertoire « list owner »).
# Les listes de propriétaires de listes de diffusion peuvent être simplement crées en
# créant un fichier dans le répertoire /etc/smail/lists/owner. Ces listes permettent
# de diffuser localement les messages d'erreurs générées par une liste de diffusion.
# Pour créer une liste des propriétaires d'une liste de diffusion,
# créer un fichier portant le nom de la liste dans /etc/smail/lists/owners.
# Cela va créer une adresse de courrier owner-nom_de_liste,
# telle qu'utilisée par le directeur « lists » plus haut.
owners: driver = forwardfile,
        caution,                        # marque toutes les adresses avec l'indicateur « caution »,
        nobody,                         # puis les associent à l'utilisateur nobody ;
        owner = postmaster;
        prefix = "owner-",
        file = lists/owner/${lc:user}   # les listes sont sous $smail_lib_dir.

# request - développe les listes de diffusions stockées dans le répertoire des listes
# de requêtes. Les listes de diffusion des requêtes concernant une liste de diffusion
# peuvent être crées tout simplement en créant un fichier dans le répertoire
# /etc/smail/lists/request. Les adresses requête sont utilisées en général
# comme adresse standard pour les demandes concernant une liste de diffusion.
# Par exemple, les demandes d'abonnement ou de désabonnement pour une liste seront
# en général envoyées à l'adresse « nom_de_liste-request », qui devra être configurée
# pour faire suivre le courrier à la ou les personnes appropriées.
request: driver = forwardfile,
        caution,                        # marque toutes les adresses avec l'indicateur « caution »,

        nobody,                         # puis les associent à l'utilisateur nobody ;
        owner = postmaster;
        suffix = "-request",
        file = lists/request/${lc:user} # les listes sont sous $smail_lib_dir.
```

Vous de devriez pas avoir besoin de changer quoi que ce soit ici, seulement les option destinées aux listes de diffusions si vous voulez gérer de telles listes avec smail, ou bien les options de renvoi (*forward options*) si, par exemple vous voulez interdire les renvois.

Le fichier `fidopaths`

```
.f105.n324.z2.fidonet.org       f105.n324.z2.fidonet.org!%s
.n324.z2.fidonet.org            f105.n324.z2.fidonet.org!%s
.z2.fidonet.org                 f105.n324.z2.fidonet.org!%s
.fidonet.org                    f105.n324.z2.fidonet.org!%s
```

Créez ce fichier seulement si vous utilisez ifmail et FIDO.

Le fichier ***routers***

```
# forces - force l'utilisation d'un chemin de distribution précis
# L'existence de cette base de donnée est un moyen de coder en dur les chemins d'accès de
# diverses machines ou domaines. Utilisé pour créer des distorsions temporaires
# des autres bases de données de routage. Pour changer la base, éditer le
# fichier maps/force.path puis faire un « make » dans le sous-répertoire maps/
forces:
        driver = pathalias,             # nom du routeur effectuant les recherches
                                        #   dans les bases de chemins ;
        method = /etc/smail/maps/table; # les transports sont dans ce fichier ;
        file = forcepaths,              # fichier contenant l'information sur
                                        #   les chemins forcées ;
        proto = lsearch,                # utiliser le fichier de chemins trié ;
        optional,
        reopen                          # se fermer lorsqu'il n'est pas utilisé.

uucp_neighbors:
        driver=uuname,                  # utiliser un programme renvoyant les voisins ;
        transport=uux;
        cmd="/usr/bin/uuname -a",       # plus précisément, utiliser le programme uuname ;
#        domain=uucp                    # enlever le suffixe «~.uucp~»

# smart_host - un directeur vers une «~machine de relais postal~» partiellement définie
# Si l'attribut smart_path du fichier de configuration contient un
# chemin entre l'hôte local et un hôte distant, alors les noms d'hôtes
# qui ne seraient pas reconnus autrement seront réexpédiés vers l'hôte distant
# précédemment mentionné (la machine de relais postal). L'attribut du fichier
# de configuration smart_transport peut être utilisé pour indiquer un moyen
# de transport différent vers la machine-relais.
# Si l'attribut smart_path n'est pas spécifié, ce routeur est ignoré.
smart_host:
        driver = smarthost,             # Pilote conçu pour ce cas spécial
        transport = uux                 # Par défaut, distribuer via UUCP
#       path=phreak

# ifmail - pour envoyer des courriers à fidonet et vice versa.
ifmail:
        driver=pathalias,
        transport=ifmail;
        file=fidopaths,
```

```
proto=lsearch
```

Vous ne devez inclure la partie ifmail que si vous utilisez ifmail pour les courriers FIDO. Il est à noter que vous pouvez aussi changer le mode de transport de *uux* (i.e., UUCP) vers, par exemple, *smtp*, ou même coder en dur les chemins vers différentes machines ou domaines dans `/etc/smail/maps/table`.

C'est utile si vous voulez que des courriers sortants pour votre réseau local soient distribués immédiatement, puisque il n'auront pas besoin d'être routé vers la connexion UUCP de votre accès internet.

Le fichier `transports`

```
# local - distribution du courrier aux utilisateurs locaux
# Indique à smail d'ajouter directement le courrier à la fin des fichiers boîtes
# aux lettres des utilisateurs dans le répertoire /var/spool/mail

#local: driver = appendfile,           # ajoute le message à la fin d'un fichier ;
#       -return_path,                  # ajoute le champ « Return-Path: » ;
#       local,                         # utiliser la forme locale de distribution ;
#       from,                          # ajoute la ligne d'enveloppe « From_ » ;
#       unix_from_hack;                # insère > avant From dans le corps du message ;
#
#       file = /var/spool/mail/${lc:user},      # utiliser cet emplacement pour Linux ;
#                                               # à noter, le répertoire de stockage de
                                                #   courrier doit avoir des droits à 1777 ;
#       file = ~/mailfile,      # utiliser cet emplacement pour une sécurité accrue ;
#       group = mail,           # groupe devant être propriétaire des fichiers pour le System V ;
#       mode = 0660,            # sous System V, le groupe mail doit avoir les droits d'accès ;
#       suffix = "\n",          # ajouter d'une ligne supplémentaire.
#       append_as_user,

# Ceci permet à chaque utilisateur d'avoir un fichier ~/.procmailrc pour contrôler
# le filtrage du courrier et permettre de sauvegarder les courriers provenant de
# listes de diffusions dans des boîtes aux lettres séparées s'ils le souhaitent.

local:  +inet,
        -uucp,
        driver = pipe,                 # envoi du message via un tube
        return_path,                   # inclus un champ « Return-Path: » ;
        local,                         # utiliser la forme locale de distribution ;
        from,                          # ajoute la ligne d'enveloppe « From_ » ;
        unix_from_hack;                # insère > avant From dans le corps du message ;

        cmd = "/usr/bin/procmail",     # utilise procmail pour distribuer le courrier local ;
        parent_env,                    # tire les informations d'environnement de
                                       #   l'adresse du parent ;
        pipe_as_user,                  # utilise l'identifiant utilisateur
                                       #   associé avec l'adresse ;
        umask = 0022,                  # droits par défaut du processus fils ;
#       -ignore_status,                # il faut faire confiance au code de retour ;
#       -ignore_write_errors,          # réessayer lorsqu'un tube est brisé.

# pipe - distribue le courrier à des commandes shells
# Ceci est implicitement utilisé quand smail rencontre des adresses commençant
# par le caractère barre verticale, comme « |/usr/lib/news/recnews talk.bizarre ».
# La barre verticale est enlevée de l'adresse avant d'être transmise au transport.
```

```
#pipe:  driver = pipe,                  # envoyer le message à un autre programme
#                                       #   via un tube ;
#       return_path, local, from, unix_from_hack;
#
#       cmd = "/bin/sh -c $user",       # envoyer l'adresse au shell Bourne ;
#       parent_env,                     # tire les informations d'environnement de
                                        #   l'adresse du parent ;
#       pipe_as_user,                   # utilise l'identifiant utilisateur
                                        #   associé avec l'adresse ;
#       umask = 0022,                   # droits par défaut du processus fils ;
#       -log_output,                    # ne pas enregistrer stdout/stderr dans les logs ;
#       ignore_status,                  # le code de retour peut être faux, l'ignorer ;
#       ignore_write_errors,            # ignorer les tubes brisés.

# file - distribue le courrier dans des fichiers
# Utilisé implicitement lorsque smail rencontre des adresses qui commencent
# par un barre de division « / » ou un tilde « ~ », comme « /usr/info/list_messages »
# ou peut être « ~/Mail/inbox ».

#file:  driver = appendfile,
#       return_path, local, from, unix_from_hack;
#
#       file = $user,                   # le nom du fichier est pris dans l'adresse ;
#       append_as_user,                 # utilise l'identifiant utilisateur
                                        #   associé avec l'adresse ;
#       expand_user,                    # développe ~ et $ dans l'adresse.
#       check_path,
#       suffix = "\n",
#       mode = 0644

# uux - distribue le courrier au programme rmail sur un site UUCP distant
#
# Il est possible, au cours d'une seule transaction UUCP, de distribuer
# le courrier destiné à 5 adresses électroniques.

uux:    driver = pipe,
        -uucp,
        inet,
#       uucp,                           # utilise le format d'adresse de style UUCP ;
        from,                           # fourni une ligne d'enveloppe « From_ » ;
        max_addrs = 5,                  # au plus 5 adresses par appel ;
        max_chars = 200;                # au plus 200 caractères pour une adresses ;

# l'option -r évite une distribution immédiate, les parenthèses autour de la variable
# $user empêche qu'elle soit interprêté par uux.

        cmd = "/usr/bin/uux - -r -g$grade $host!rmail $(($[strip:user})$)",
#        cmd="/usr/bin/uux - $host!rmail $(($user)$)",
        ignore_write_errors,            # ignore les tubes brisés.
        umask = 0022,
#       pipe_as_sender,

# uux_one_addr - distribue le courrier par UUCP à un hôte distant qui ne peut
#                accepter qu'une seule adresse à la fois.
```

```
#
# Cela est souvent nécessaire quand on distribue du courrier à un site qui
# tourne sous une version non modifiée de 4.1BSD.

uux_one_addr:
        driver = pipe,
        uucp,                           # utilise le format d'adresse de style UUCP ;
        from;                           # fourni une ligne d'enveloppe « From_ » ;

        # l'option -r empêche une distribution immédiate.

        cmd = "/usr/bin/uux - -r -g$grade $host!rmail (${strip:user})",
        umask = 0022,
        pipe_as_sender

queueonly:
        driver = pipe;                  # envoi du message via un tube ;
        cmd = "/usr/lib/sendmail -Q -f $sender -bm $user",
                                        # utilise getmail pour les distributions locales ;
        user=root,                      # exécute getmail en tant qu'utilisateur « root » ;
        group=mail,                     # exécute getmail sous le groupe « mail » ;
        parent_env,                     # tire les informations d'environnement de
                                        #   l'adresse du parent ;
        -pipe_as_user,                  # utilise l'identifiant utilisateur
                                        #   associé avec l'adresse ;
        umask = 0007,                   # droits par défaut du processus fils.

# pour distribuer un message. Le transport SMTP n'est inclus que si le support
# réseau BSD est présent.
# L'attribut uucp peut être positionné pour le transfert dans une zone uucp.
# L'attribut inet doit être positionné pour les transferts vers l'internet.
# note : c'est loin d'être optimal, un logiciel d'arrière-plan devrait exister
# qui puisse prendre en charge plusieurs messages par connection.
# de plus : il peut être nécessaire de restreindre max_addrs à 100, puisqu'il
# s'agit de la limite inférieure qu'une mise en oeuvre de SMTP est tenue de
# prendre en charge.

smtp:   driver=tcpsmtp,
        inet,                           # si UUCP_ZONE n'est pas défini ;
#       uucp,                           # si UUCP_ZONE est défini ;
        -max_addrs, -max_chars;         # pas de limite sur le nombre d'adresses ;

        short_timeout=5m,               # la durée maximum des opérations courtes ;
        long_timeout=2h,                # la durée maximum des opérations SMTP plus longues ;
        service=smtp,                   # se connecter à ce port de service ;
# Pour l'utilisation internet : ne pas commenter les 4 lignes qui suivent
       use_bind,                        # résoud les enregistrements A multiples et MX ;
       defnames,                        # utiliser la recherche standard de domaines ;
       defer_no_connect,                # essayer à nouveau si le serveur est tombé ;
       local_mx_okay,                   # évite un MX vers l'hôte local.

ifmail:
        from,received,max_addrs=5,max_chars=200,
        driver=pipe;
        pipe_as_sender,
```

```
        cmd="/usr/local/bin/ifmail -x9 -r$host $(${strip:user})$)"
```

Vous ne devez inclure la partie ifmail que si vous utilisez ifmail pour les courriers FIDO. À part cela, vous ne devriez pas avoir besoin de modifier quoi que ce soit dans ce fichier qui définit les agents de transport (comme UUX, SMTP, etc.) que vous pouvez utiliser comme paramètres dans les autres fichiers de configuration.

Remarquez que j'ai commenté quelques parties comme *pipes* ou *file* pour améliorer la sécurité.

Le répertoire `maps` Il contient les fichiers `map` et `table`.

Tout d'abord le fichier `map` :

```
#N      foo.bar foo2.bar2
#S      AT 486/RedHat Linux 1.2.13
#O      organization
#C      contact
#E      administration (email)
#T      phone
#P      address
#R
#U      hosts connected via uucp
#W      created/edited by
#
hname polux

hname linux.eu.org

hname = polux
hname = polux.linux.eu.org
```

Encore une fois, adaptez le fichier à votre situation (je suis alimenté par `polux.linux.eu.org`).

Maintenant le fichier `table` :

```
*       uux
```

Vous pouvez définir différents transports pour différents chemins, par exemple *smtp* pour les machines de votre réseau local, *uux* pour le reste du monde ou vice-versa (j'utilise UUCP pour tous les courriers sortants, donc j'utilise * !)

6.2.2 Autres bons exemples

Les fichiers précédents sont ceux que j'utilise pour mon site ; vous ne devriez pas rencontrer de problèmes en les utilisant comme bases pour vos propres fichiers.

Les fichiers suivants sont donnés comme de bons exemples pour configurer smail de différentes manières.

```
#ident "@(#) transports,v 1.2 1990/10/24 05:20:46 tron Exp"

  # Voir smail(5) pour une description complète du contenu de ce fichier

  # local - distribution du courrier aux utilisateurs locaux
  #
```

```
# Indique à smail d'ajouter directement le courrier à la fin des fichiers boîtes
# aux lettres des utilisateurs, contenus dans le répertoire « /usr/mail ».
local:  driver = appendfile,            # ajouter le message à la fin du fichier ;
        return_path,                    # ajoute le champ « Return-Path: » ;
        local,                          # utiliser la forme locale de distribution ;
        from,                           # ajoute la ligne d'enveloppe « From_ » ;
        unix_from_hack;                 # insère > avant From dans le corps du message ;

        file = /usr/mail/${lc:user},    # utiliser cet emplacement pour System V ;
        group = mail,                   # groupe devant être propriétaire des fichiers
                                        #   pour le System V ;
        mode = 0660,                    # sous System V, le groupe mail doit avoir les
                                        #   droits d'accès ;
        suffix = "\n",                  # ajouter d'une ligne supplémentaire.
        append_as_user,

# pipe - distribue le courrier à des commandes shells
#
# Ceci est implicitement utilisé quand smail rencontre des adresses commençant
# par le caractère barre verticale, comme « |/usr/lib/news/recnews talk.bizarre ».
# La barre verticale est enlevée de l'adresse avant d'être transmise au transport.

pipe:   driver = pipe,                  # envoyer le message à un autre programme
                                        #   via un tube ;
        return_path, local, from, unix_from_hack;

        cmd = "/bin/sh -c $user",       # envoyer l'adresse au shell Bourne ;
        parent_env,                     # tire les informations d'environnement de
                                        #   l'adresse du parent ;
        pipe_as_user,                   # utilise l'identifiant utilisateur
                                        #   associé avec l'adresse ;
        umask = 0022,                   # droits par défaut du processus fils ;
        -log_output,                    # ne pas enregistrer stdout/stderr ;
        ignore_status,                  # le code de retour n'est pas fiable, l'ignorer ;
        ignore_write_errors,            # ignorer les tubes brisés.


# file - distribue le courrier dans des fichiers
#
# Utilisé implicitement lorsque smail rencontre des adresses qui commencent
# par un barre de division « / » ou un tilde « ~ », comme « /usr/info/list_messages »
# ou peut être « ~/Mail/inbox ».

file:   driver = appendfile,
        return_path, local, from, unix_from_hack;

        file = $user,                   # le nom de fichier est pris dans l'adresse
        append_as_user,                 # utilise l'identifiant utilisateur
                                        #   associé avec l'adresse ;
        expand_user,                    # développe ~ et $ dans l'adresse.
        suffix = "\n",
        mode = 0644

# uux - distribue le courrier au programme rmail sur un site UUCP distant
#
```

```
# Il est possible, au cours d'une seule transaction UUCP, de distribuer
# le courrier destiné à 5 adresses électroniques.

uux:    driver = pipe,
        uucp,                           # utilise le format d'adresse de style UUCP ;
        from,                           # fourni une ligne d'enveloppe « From_ » ;
        max_addrs = 5,                  # au plus 5 adresses par appel ;
        max_chars = 200;                # au plus 200 caractères pour une adresse ;

        # l'option -r évite une distribution immédiate, les parenthèses autour de la variable
         # $user empêche qu'elle soit interprêté par uux.

        cmd = "/usr/bin/uux - -r -g$grade $host!rmail $(($${strip:user})$)",
        umask = 0022,
        pipe_as_sender


# uux_one_addr - distribue le courrier par UUCP à un hôte distant qui ne peut
#                accepter qu'une seule adresse à la fois.
#
# Cela est souvent nécessaire quand on distribue du courrier à un site qui
# tourne sous une version non modifiée de 4.1BSD.

uux_one_addr:
        driver = pipe,
        uucp,                           # utilise le format d'adresse de style UUCP ;
        from;                           # fourni une ligne d'enveloppe « From_ » ;

        # l'option -r empêche une distribution immédiate ;

        cmd = "/usr/bin/uux - -r -g$grade $host!rmail ($${strip:user})",
        umask = 0022, pipe_as_sender

# demand - distribution vers un programme rmail distant, connexion à la demande

demand: driver = pipe,
        uucp, from, max_addrs = 5, max_chars = 200;

        # si l'option -r est omise, essaye de contacter le site distant immédiatement.
        cmd = "/usr/bin/uux - -g$grade $host!rmail $(($user)$)",
        umask = 0022, pipe_as_sender

# uusmtp - distribution vers un programme rsmtp sur un site UUCP distant
#
# Distribution via un simple protocol de transfert SMTP par lots
# vers la machine distante.
# Ce qui permet d'utiliser des adresses bien plus libres et qui évite les
# limitations de uux en termes de nombre d'adresses de destination.

uusmtp: driver = pipe,
        bsmtp,                          # envoi des lots de commandes SMTP ;
        -max_addrs,                     # il n'y a pas de limite sur le nombre
        -max_chars;                     # et la taille des adresses de destination ;

  # mettre -r pour que la distribution ne soit pas immédiate ; les adresses de
```

```
  # destination sont stockées dans les données envoyées vers l'entrée standard de rsmtp.

        cmd = "/usr/bin/uux - -r -g$grade $host!rsmtp",
        umask = 0022, pipe_as_sender

# demand_uusmtp - distribution vers un programme rsmtp distant, connexion à la demande

demand_uusmtp:
        driver = pipe,
        bsmtp, -max_addrs, -max_chars;

        # si l'option -r est omise, essaye de contacter le site distant immédiatement.
        cmd = "/usr/bin/uux - -g$grade $host!rsmtp",
        umask = 0022, pipe_as_sender

# smtp - distribution du courrier en utilisant SMTP sur TCP/IP
#
# Se connecte à un hôte distant via TCP/IP et initie une conversation SMTP pour
# distribuer un message.
# Le transport SMTP n'est inclus que si le support réseau BSD est présent.


# note : il peut être nécessaire de restreindre max_addrs à 100, puisqu'il
# s'agit de la limite inférieure qu'une mise en oeuvre de SMTP est tenue de
# prendre en charge.

smtp:   driver = smtp,
        -max_addrs,
        -max_chars

#ident "@(#) table,v 1.2 1990/10/24 05:20:31 tron Exp"

# Ce fichier indique les transports utilisés pour distribuer le courrier
# à des hôtes spécifiques à partir de bargw.

#host           transport
#--------       ---------
curdsgw         demand_uusmtp   # distribue le courrier via des lots de connexions SMTP ;
oldbsd          uux_one_addr    # les sites 4.1BSD ne peuvent accepter plus d'une adresse ;
sun             demand          # appelle « sun » lorsqu'il y a du courrier à envoyer ;
*               uux             # pour tous les autres, connexion à intervalles réguliers.
```

6.2.3 Relancer inetd

Pour lancer smail en tant que démon SMTP, ajoutez une des lignes suivantes dans votre `/etc/inetd.conf` :

```
smtp stream tcp nowait  root  /usr/bin/smtpd smtpd
```

ou :

```
smtp stream tcp nowait  root  /usr/sbin/tcpd  /usr/sbin/in.smtpd
```

Le courrier sortant sera alors envoyé automatiquement quand vous utiliserez elm.

6.2.4 Smail et SMTP

Généralement les FAI utilisent SMTP. Par conséquent, vous ne devriez pas avoir de problème pour envoyer votre courrier. Quand vous n'êtes pas connecté à Internet, lorsque vous envoyez un courrier électronique, il est stocké dans `/var/spool/mail/input`. Ensuite, quand vous vous connecterez, *runq* sera lancé et votre courrier envoyé. Cependant, le vrai problème est la réception de votre courrier puisque votre FAI doit s'occuper de plusieurs clients, et non pas uniquement de vous !

Généralement, vous pouvez récupérer votre courrier via le protocole POP (reportez-vous à la section POP plus bas).

6.3 Sendmail+IDA (attention, cette section n'est pas à jour)

Pour les gros sites, sendmail est un choix valable, grâce à son « incroyable facilité d'utilisation » (sentiment très relatif quand on connaît qmail), mais vous devrez choisir entre sendmail+IDA et sendmail 8.x :

- Si vous utilisez un vieux noyau (1.0) : sendmail+IDA.
- Si vous utilisez un noyau un peu moins vieux (1.2) : sendmail+IDA et la modification du code source
- Ceux qui utilisent un noyau récent (2.0) choisirons sendmail 8.x .

Attention, les linuxiens débutants et les personnes préoccupées par la sécurité et la simplicité de configuration devraient plutôt essayer smail ou qmail, qui sont plus faciles à utiliser et plus sécurisés.

6.3.1 Installation à partir des sources

Si votre distribution n'est pas fournie avec un paquetage sendmail prêt-à-l'emploi (.rpm pour RedHat, Caldera et SuSE, .deb pour Debian) téléchargez les sources et lancez :

- cd / ; tar -zxvf sendmail5.67b+IDA1.5.tgz
- Placez vous dans le répertoire `/usr/local/lib/mail/CF` et concaténez les fichiers `sample.m4` et `local.m4` dans *votre_nom_de_machine*`.m4`.

Éditez *hostname*, *aliases* et *smarthost* et placez-y les informations correspondant à votre site. Le fichier par défaut est prévu pour un site utilisant uniquement UUCP (ce qui n'est plus le cas des versions 8.x), qui a des entêtes de domaine et qui dialogue avec une machine « de relais postal » (*smarthost*). Ensuite lancez `make votre_nom_de_machine.cf` et copiez le fichier créé dans `/etc/sendmail.cf`.

Si vous utilisez seulement UUCP, vous n'avez besoin de créer *aucune* des tables mentionnées dans le fichier `README.linux`. Vous n'aurez qu'à utiliser *touch* sur les fichiers pour que le `Makefile` marche. Éditez ensuite le fichier .m4, lancez `make sendmail.cf` et testez-le.

Si votre site utilise uniquement UUCP et que vous dialoguez avec des sites autres que votre machine-relais, vous devrez ajouter des entrées uupcpxtable pour chacun d'entre eux (sinon le courrier qui leur sera envoyé passera par la machine-relais) et lancer dbm sur la nouvelle uucpxtable.

Si vous utilisez la distribution binaire de Rich Braum de la version 5.67a, vous devrez lancer `/usr/lib/sendmail -bz` pour que les changements soient pris en compte.

Vous devriez également mettre à jour votre version vers au moins la 5.67b puisque qu'il y a un méchant trou de sécurité dans la version 5.67a et les précédentes. Un autre point sympathique est que si vous activez l'option mail.debug et que vous utilisez syslogd, la liste de vos courriers entrants et sortants sera enregistrée. Voyez le fichier `/etc/syslogd.conf/` pour plus de détails.

Les sources de sendmail+IDA se trouve sur `http://vixen.cso.uiuc.edu` ; elles ne nécessitent aucun correctif pour tourner sous Linux si vous utilisez quelque-chose comme un noyau 1.00.

Si vous utilisez un noyau dont la version est supérieur à la 1.1.50, vous allez devoir vous amuser à enlever tous les correctifs spécifiques à Linux qui se trouvent dans les sources officielles. (Je vous avais prévenu que ce sendmail était seulement pour les vieux noyaux :-)

Il est extrêmement facile de repérer les endroits à corriger : lancez *make* et quand il vous hurle dessus, allez à la ligne du fichier qu'il vous indique et commentez le code spécifique à Linux qui s'y trouve.

Si vous avez opté pour sendmail+IDA, je vous recommande fortement d'utiliser la version sendmail5.67b+IDA1.5, car tous les correctifs spécifiques à Linux sont maintenant dans les sources officielles et plusieurs trous de sécurité qui *étaient* dans les anciennes versions que vous auriez récupérées ou compilées avant le 1er décembre 1993 ont été réparés.

Maintenant que l'on en est au noyau 2.0 (NDT : et même au 2.4 au moment où j'écris ces lignes) vous devriez utiliser sendmail 8.x à la place de sendmail+IDA, mais je vous ai déjà dit que vous devriez choisir sendmail 8.x :-)

6.3.2 Le fichier `sendmail.m4`

Sendmail+IDA se configure par l'intermédiaire du fichier `sendmail.m4` et non directement via le fichier `sendmail.cf`. Ce qui rend cette façon de travailler intéressante est que grâce à elle, il est simple de mettre en place des configurations qui seraient extrêmement difficiles (voire totalement impossibles) à réaliser avec smail ou avec le sendmail traditionnel.

Le fichier `sendmail.m4` qui correspond à la configuration smail que nous avons décrite au-dessus ressemble à ceci :

```
dnl #------------------- EXEMPLE DE FICHIER SENDMAIL.M4 -------------------
dnl #
dnl # La chaîne « dnl » est utilisée comme en-tête d'une ligne de commentaires.
dnl # (Enfin pas exactement, mais utilisez-la pour cela si vous en avez
dnl #  besoin :-)
dnl # En général, il est préférable d'éviter de remplacer les chemins par
dnl # défaut dans LIBDIR
dnl #define(LIBDIR,/usr/local/lib/mail)dnl   # dossier contenant tous les
dnl                                         # fichiers de support~;
define(LOCAL_MAILER_DEF, mailers.linux)dnl  # outil de distribution locale~;
define(POSTMASTERBOUNCE)dnl                 # les messages retournés sont
dnl                                         # envoyés au «~receveur principal~»
dnl                                         # (postmaster)~;
define(PSEUDODOMAINS, BITNET UUCP)dnl       # ne pas essayer le DNS sur ceux-ci.
dnl #
dnl #-----------------------------------------------------------
dnl #
dnl # noms par lesquels nous sommes connus
define(PSEUDONYMS, machine.sousdomaine.domaine machine.UUCP)
dnl #
dnl # notre nom de machine
define(HOSTNAME, machine.sousdomaine.domaine)
```

```
dnl #
dnl # notre nom UUCP
define(UUCPNAME, machine)dnl
dnl #
dnl #------------------------------------------------------------
dnl #
define(UUCPNODES, |uuname|sort|uniq)dnl       # nos voisins UUCP~;
define(BANGIMPLIESUUCP)dnl                    # s'assure que le courrier
define(BANGONLYUUCP)dnl                       # UUCP est correctement traité~;
define(RELAY_HOST, mon_voisin_UUCP)dnl        # notre machine-relais~;
define(RELAY_MAILER, UUCP-A)dnl               # on atteind notre machine-
dnl                                           # relais par UUCP
dnl #
dnl #-------------------------------------------------------------------
dnl #
dnl # les diffentes tables dbm
dnl #
define(ALIASES, LIBDIR/aliases)dnl            # les alias système
define(DOMAINTABLE, LIBDIR/domaintable)dnl    # machines du domaine
define(PATHTABLE, LIBDIR/pathtable)dnl        # base de données des chemins
define(GENERICFROM, LIBDIR/generics)dnl       # adresses sources génériques
define(MAILERTABLE, LIBDIR/mailertable)dnl    # agents de courrier par machine
dnl                                           # ou par domaine
define(UUCPXTABLE, LIBDIR/uucpxtable)dnl      # chemin des machines que l'on
dnl                                           # alimente
define(UUCPRELAYS, LIBDIR/uucprelays)dnl      # chemins rapides
dnl #
dnl #-------------------------------------------------------------------
dnl #
dnl # inclut le «~véritable~» code qui fait tout marcher
dnl # (fournit avec le code source)
dnl #
include(Sendmail.mc)dnl                         # entrée nécessaire !
dnl #
dnl #------------ FIN DE L'EXEMPLE DE FICHIER SENDMAIL.M4 ------- -------
```

6.3.3 Choisir un logiciel de distribution locale du courrier

À la différence de la plupart des distributions Unix, Linux n'est pas fourni avec un agent par défaut de distribution du courrier local.

La distribution Slackware est l'exception qui confirme la règle. Du moins, cette fonctionnalité est proposé par le script d'installation facile-à-utiliser-mais-de-longue-haleine. C'est procmail qui est utilisé.

Actuellement, *deliver* ou *procmail* sont en général installés, avec une configuration par défaut de sendmail adaptée à la distribution du courrier local. De sorte qu'aucune difficulté additionnelle ne vient pimenter une configuration déjà très complexe. Je recommande d'utiliser *deliver* ou *procmail*, qui sont très faciles à trouver, et qui peuvent être proposés sous la forme de paquetages optionnels dans quelques distributions Linux.

Dans ce but, vous aurez besoin de définir *LOCAL_MAILER_DEF* dans le fichier *sendmail.m4* qui pointe vers un fichier de ce genre :

```
# -- /usr/local/lib/mail/mailers.linux --
#     (agents de distribution locaux pour Linux)
Mlocal, P=/usr/bin/deliver, F=SlsmFDMP, S=10, R=25/10, A=deliver $u
Mprog,  P=/bin/sh,       F=lsDFMeuP,   S=10, R=10, A=sh -c $u
```

Il y aussi une option par défaut d'utilisation de *deliver* incluse dans le fichier `Sendmail.mc` qui est importé par le fichier `sendmail.cf`. Pour vous en servir, n'utilisez pas le fichier *mailers.linux* mais incluez ce qui suit dans votre fichier `sendmail.m4` :

```
dnl --- (dans sendmail.m4) ---
define(LOCAL_MAILER_DEF, DELIVER)dnl    # agent de distribution locale
```

Malheureusement, `Sendmail.mc` considère que *deliver* est installé dans `/bin` ce qui n'est pas le cas sur une Slackware1.1.1 (il est installé dans `/usr/bin`). Dans ce cas, vous aurez besoin soit de créer un lien, soit de reconstruire *deliver* à partir des sources pour qu'il réside dans `/bin`. À noter que *procmail* est généralement meilleur que *deliver*, par exemple pour le filtrage du courrier.

6.3.4 Tables dbm de sendmail+IDA

La mise en place de comportements spéciaux vers certains sites et domaines est réalisée via un certain nombre de tables *dbm* plutôt que par une modification directe du fichier `sendmail.cf`.

Consultez le numéro de juillet 1994 du *Linux Journal* (si vous pouvez encore le trouver :-) et référez-vous aux docs qui se trouvent dans les sources, ou au chapitre sendmail de la nouvelle version de *Networking Administration Guide* du *Linux Documentation Project* qui sera disponible incessamment sous peu.

- mailertable : définit un comportement spécial pour les machines et les domaines distants.
- uucpxtable : force la distribution de courrier via UUCP pour des machines qui sont au format DNS.
- pathtable : définit le *bang-paths* UUCP utilisé vers les machines ou domaines distants.
- uucprelays : court-circuite le chemin *pathalias* vers des machines distantes connues.
- genericfrom : convertit les adresses internes en adresses génériques visibles par le monde extérieur.
- xaliases : convertit les adresses génériques vers ou à partir des adresses internes.
- decnetxtable : convertit les adresses RFC-822 en adresses DECnet.

6.3.5 Quelles sont les entrées réellement nécessaires ?

Quand il n'utilise aucune table dbm optionnelle, sendmail distribue le courrier via le *RELAY_HOST* (machine de relais) et le *RELAY_MAILER* (agent de relais) défini dans le fichier `sendmail.m4` utilisé pour générer `sendmail.cf`. Il est facile de redéfinir ce comportement via des entées dans `domaintable` ou `uucpxtable`.

Un site générique qui est sur Internet et utilise DNS, ou qui n'utilise qu'UUCP et fait suivre le courrier via UUCP par une machine de relais postal, n'a probablement besoin d'aucune entrées dans ces tables.

Virtuellement, tous les systèmes devraient définir les macrosi *DEFAULT_HOST* et *PSEUDODYMS* , qui définissent le nom canonique du site et les alias par lesquels il est connu.

Si vous n'avez qu'une machine de relais et un agent relais de courrier vous n'avez pas besoin de définir ces macros puisque cela marche automagiquement. Les machines UUCP nécessiteront probablement de définir *UUCPNAME* comme leur nom officiel UUCP.

Vous devrez probablement également définir *RELAY_MAILER* et *RELAY_HOST* qui active le routage via une machine de relais postal.

Le type de transport du courrier utilisé doit être défini dans *RELAY_MAILER*. Il devrait être en général *UUCP-A* pour les sites UUCP sites. Si votre site n'utilise que SMTP et utilise un DNS, vous devrez changer *RELAY_MAILER*.

Si vous utilisez SLIP, la manière la plus simple de configurer votre site est de faire suivre tout votre courrier sortant vers votre fournisseur d'accès à Internet. Pour cela, définissez les macros *ISOLATED_DOMAINS* et *VALIDATION_DOMAINS* pour qu'elles correspondent à votre nom de domaine. Définissez également *RELAY_HOST* qui doit contenir le nom de la machine de votre fournisseur d'accès à Internet, et *RELAY_MAILER* qui doit contenir *TCP*. Bien sûr, vous devez obtenir la permission avant d'utiliser une machine en tant que machine de relais principale.

6.4 Sendmail 8.x

Sendmail 8.7.x de Berkeley a été la dernière version importante avant Sendmail5. Elle se compilait et se configurait très simplement sous Linux en tapant : `make linux`.

Vous feriez probablement mieux de vous procurer une des différentes distributions binaires sur un site d'archive dédié à Linux plutôt que des vous battre avec des choses comme Berkeley dbm.

Il existe une bonne distribution de sendmail 8.6.12 de Jason Haar (<`j.haar at lazerjem.demon.co.uk`>) sur *sunsite.unc.edu* `/pub/Linux/system/Mail/delivery/sendmail-8.6.12-bin.tgz` qui contient les documentations d'origine et une petite documentation très sympa qui explique comment utiliser sendmail v8 sur les principales configurations.

L'idée principale de sendmail v8 est que l'on veut configurer le strict minimum pour que cela marche. Ce qui suit est un exemple qui devrait vous permettre de vous rapprocher de cela.

6.4.1 Un exemple de fichier mc 8.7.x

Exactement comme sendmail+IDA, sendmail v8 utilise m4 pour générer à partir d'un fichier de configuration le fichier `sendmail.cf` complet utilisé par `sendmail`. Voici le fichier mc que j'utilise pour mon site (ppp vers Internet pour le courrier sortant et UUCP pour le courrier rentrant).

```
dnl divert(-1)
#-----------------------------------------------------------------
#
# Voici le fichier .mc pour une machine linux configurée comme ceci :
#
#       - connecté via ppp pour le courrier sortant
#       - connecté via UUCP pour le courrier entrant
#       - en-tête avec domaine
#       - aucun agent de courrier local (utilise deliver à la place)
#       - sans DNS local et donc n'utilise pas le DNS pour transformer
#         les adresses des courriers sortants en leurs formes canoniques
#       - tous les courriers sortants qui ne sont pas locaux passent
#         par la machine-relais (RELAY_HOST) via SMTP (on utilise ppp
#         et on laisse notre FAI faire son boulot)
```

```
#
#                                       vds 3/31/95
#
#---------------------------------------------------------------------
include(`../m4/cf.m4')
VERSIONID(`machine Linux sans DNS transferant son courrier vers la'dnl
`machine-relais d'un FAI via slip')dnl
Cwmachine.mon.domaine machine.UUCP localhost
OSTYPE(linux)
FEATURE(nodns)dnl
FEATURE(always_add_domain)dnl
FEATURE(redirect)
FEATURE(nocanonify)
dnl MAILER(local)dnl
MAILER(smtp)dnl
MAILER(uucp)dnl
define(`RELAY_HOST', smtp:relai.machine.domaine)
define(`SMART_HOST', smtp:relai.machine.domaine)
define(`UUCP_RELAY', smtp:relai.machine.domaine)
define(`LOCAL_MAILER_PATH', `/bin/deliver')
define(`LOCAL_MAILER_ARGS', `deliver $u')
```

6.4.2 Les bons morceaux de Sendmail v8

Il y a quelques différences, je suppose, pour les fanatiques d'IDA.

Jusqu'ici j'ai trouvé la suivante :

À la place de `runq`, tapez `sendmail -q` pour traiter la file d'attente.

6.5 Agents de distribution du courrier local (LDA)

À la différence de la plupart des systèmes d'exploitation, Linux ne gère pas le courrier lui-même : vous avez besoin d'un programme pour distribuer le courrier local comme par exemple *lmail*, *procmail* ou *deliver*.

Cependant, toutes les distributions récentes sont à présent fournies avec un agent de distribution local.

La documentation expliquant comment les utiliser pour la distribution locale du courrier est incluse dans la distribution binaire de sendmail5.67b+IDA1.5 (sur sunsite) mentionné ci-dessus.

7 Administration des lecteurs de courrier électronique (MUA)

7.1 Mutt

Vous ne devriez pas avoir de problèmes pour compiler, installer et lancer mutt. Les utilisateurs de qmail peuvent récupérer un correctif ou le lancer avec l'option *-f* pour lire leurs boites aux lettres locales.

Si mutt vous embête avec un « unknown terminal error » après une mise à jour de votre distribution, recompilez-le.

7.2 elm

Elm se compile, s'installe et se lance sans problème sous Linux. Pour plus d'informations, reportez-vous aux sources d'elm et aux instructions d'installation. Elm et filter ont besoin d'être en mode 2755 (groupe mail) avec /var/spool/mail en mode 775 (groupe mail).

Les utilisateurs de qmail peuvent récupérer le correctif pour utiliser les fonctionnalités bien pratiques de qmail ou lancer elm avec l'option *-f* pour lire leurs boîtes aux lettres locales.

Si vous utilisez une distribution binaire, vous aurez besoin de créer un fichier `/usr/local/lib/elm/elm.rc` pour passer outre la nom de la machine et les informations de domaine compilés en dur :

- remplacez *sous-domaine.domaine* par votre nom de domaine ;
- remplacez *machine* par le nom de votre machine (sans le nom de domaine).

```
#---------- /usr/local/lib/elm/elm.rc ------------------
#
# nom de machine non qualifié
hostname = machine
#
# nom du domaine local
hostdomain = sous-domaine.domaine
#
# nom de machine qualifié
hostfullname = machine.sous-domaine.domaine
#
#-------------------------------------------------------
```

Une chose dont vous devez être conscient est que si vous avez compilé elm pour qu'il soit capable de gérer le format MIME, vous aurez besoin d'installer metamail dans votre chemin, sinon elm ne sera pas capable de lire le courrier MIME que vous recevrez. metamail est disponible sur et bien sûr via archie.

Dans la catégorie « trop beau pour être vrai » il existe une distribution, d'Elm-2.4.24 qui sait gérer PGP. Pour l'essayer, récupérez le fichier

`ftp://ftp.viewlogic.com/pub/elm-2.4pl24pgp3.tar.gz` qui n'est rien d'autre que elm 2.4.24 avec l'ajout de PGP. Il se configure et se compile de la même manière que la version normale d'elm, ce qui veut dire que vous aurez sans doute à ajouter les correctifs mentionnés ci-dessus. Pour ce que ça coûte, je l'utilise ici et je l'aime beaucoup. Bien sûr, il doit y avoir des version plus récentes disponibles comme par exemple elm-ME+.

De temps en temps, elm plante avec un message indiquant qu'il n'arrive pas à allouer un énorme quantité d'octets de mémoire via malloc(). Pour remédier à cela, enlevez les alias généraux de courrier post-générés (aliases.dir et aliases.pag). Bien que ce point ne soit pas spécifique à Linux, il est perçu comme un bogue gênant d'elm.

Ce n'est pas un bogue d'elm, c'est une erreur dans le configuration d'elm utilisée par la personne qui a fait la distribution binaire que vous utilisez.

Elm possède, pour les alias, un format amélioré et incompatible ; ainsi, vous devrez vous assurer que le chemin utilisé par elm pour ses alias est différent de celui utilisé par sendmail ou smail. Vu le nombre impressionnant de rapport de bogues à propos de ce problème, il est fort clair qu'au moins une des distributions majeures ait dans le passé fait une erreur de configuration sur ce point. (Ceci m'a été signalé par Scot W. Stevenson : <`scot at catzen.gun.de`>.)

Le paquetage metamail actuel à besoin de csh pour quelques-uns de ses scripts. Sans csh (ou tcsh), vous aurez droit à des erreurs des plus intéressantes...

7.3 Mailx

Si vous n'avez pas mailx, économisez vos efforts et récupérez le kit mailx de la Slackware 2.1.0 ou plus, qui a une bonne implémentation de mailx5.5. Il se compile sans correctif sous linux si vous avez installé pmake.

Si quelqu'un les utilise encore, je lui recommande fortement de virer les vieux trucs edmail de la SLS 1.00 et de les remplacer par mailx.

8 Gérer le courrier distant

Cette section décrit l'utilisation de POP et d'IMAP pour gérer le courrier distant.

Parmi les alternatives, on trouve le montage via NFS des zones de stockage du courrier sur les machines clientes (Danger Will Robinson ! Est-ce que tout le monde utilise la même méthode de verrouillage ?) ou bien l'utilisation d'une passerelle courrier-vers-web (une solution assez populaire de nos jours).

8.1 Un peu d'histoire

Sur les réseaux de stations de travail, le courrier a toujours été un problème :

- Soit vous utilisez *utilisateur@ordinateur.lambda.fr* et, vous avez des problèmes quand *ordinateur* est hors service ; vous faîtes connaître les machines qui composent votre réseau à des personnes extérieures ; vous utilisez une adresse différente par ordinateur pour un même utilisateur ; et cætera.
- Soit vous utilisez une poste centrale (*mail hub*) *machine.lambda.fr* avec des règles de réécriture pour que l'on ait l'impression que tous utilisateurs envoient leur courrier depuis la même adresse quelque-soit l'ordinateur qu'ils utilisent.

Mais dans ce cas, comment les utilisateurs peuvent-ils lire leur courrier ?

En utilisant rsh avec elm ? :-)

Cela surchargerait notre centre de tri ! Une méthode serait de rediriger le courrier ou d'utiliser UUCP, SMTP, etc. mais c'est bien trop complexe.

Vinrent ensuite POP et IMAP, tout deux avec des problèmes de sécurité au début (qui sont maintenant réglés par l'utilisation ssh sur les dernières versions). Pour les utiliser, un agent de transport de courrier doit parfois être installé sur la machine de l'utilisateur (du style qmail, smail ou vmail si par exemple vous utilisez elm, l'utilisation de mozilla vous évitera cela), cependant cela simplifiera l'envoi et la réception de courrier.

8.2 Récupérer le courrier

Voici le principaux inconvénients de POP :

- Le mot de passe est envoyé en clair sur le réseau.
- Vous devez choisir l'un des nombreux lecteur de courrier qui gèrent POP (comme Pine, Emacs, Mozilla, Netscape, Mutt, IE, Pegasus, Eudora, Claris, etc.).

- Lorsqu'un utilisateur peut travailler sur de multiples machines, le fait que son courrier soit stocké sur la machine qu'il a utilisé la veille peut être une plaie.
- Certains serveurs POP (comme qpopper ou ipop2d) sur des serveur très utilisé, peuvent freiner significativement la machine. Considérez les options de contrôle (comme ne pas laisser le courrier sur le serveur) et le remplacement de votre serveur pop (par cucipop par exemple). Évitez également de lancer votre serveur via inetd.

Le problème du mot de passe peut être résolu en utilisant POP au travers d'un canal chiffré ou en utilisant les extensions APOP ou RPOP. Le problème du lecteur de courrier peut être résolu soit en changeant de lecteur (ne sous-estimez pas l'effort qu'il faut déployer pour rééduquer les utilisateurs), soit en utilisant un programme qui rapatrie le courrier via POP (comme par exemple fetchmail) associé à un agent de distribution local.

IMAP peut être préférable à POP dans diverses situations comme par exemple si vous permettez les accès distants (et spécialement pour des utilisateurs se connectant de machines différentes). Vous pouvez également utiliser POP en restreignant son accès à un réseau local sur lequel le vol des mots de passe transitant par le réseau n'est pas à craindre. Mark Aitchison nous a signalé qu'une façon de faire cela était d'utiliser les fichiers `hosts.deny` et `hosts.allow` et de lancer votre serveur POP à partir d'inetd (reportez-vous pour de plus amples détails au « Net-3 HOWTO »).

Le choix de laisser (ou non) le courrier sur le serveur a des conséquences en termes d'espace disque, de facilité de mise en oeuvre des sauvegardes et de la sécurité, ainsi que sur la possibilité pour un utilisateur donné de se connecter depuis plusieurs machines distinctes. Ainsi la meilleure solution dépend de votre type d'organisation. Bien sûr, cela ne vous garantira pas que votre courrier ne puisse pas être lu, mais personne ne pourra l'effacer ; et si votre courrier est chiffré avec PGP, cela en fait une très bonne solution.

Voici quelques clients POP qui en valent la peine :

- gwpop (*a Good Way to POP*) est très sécurisé car il crée un canal chiffré et place le courrier directement dans l'espace de stockage ; cependant, il nécessite l'installation de Perl.
- popclient est facile à utiliser. Par exemple, si votre identifiant est *jean* et votre mot de passe est *AssezSecret*, vous lancerez

  ```
  $ popclient -3 -v mail.acme.net -u jean -p "AssezSecret" -k -o JOHN-INET-MAIL
  ```

  Ceci est fortement déconseillé sur les machines multi-utilisateurs. En effet, les autres utilisateurs pourraient voir votre mot de passe en lançant par exemple `ps auxw`.
- fetchmail, qui est activement supporté et incroyablement facile à utiliser. On le configure grâce au fichier `~.fetchmailrc`. Ainsi, vous n'aurez qu'à lancer fetchmail quand vous voudrez rapatrier votre courrier.

  Voici mon .fetchmailrc :

  ```
  poll mail.server protocol pop3:
          forcecr
          password AssezSecret;
  ```

  L'option *forcecr* est requise pour utiliser fetchmail avec qmail qui respecte strictement les RFC.

8.3 Envoyer du courrier

Pour cela, vous devrez utiliser un programme gérant SMTP comme par exemple qmail, smail, vmail ou mozilla (ce dernier fait tout : lecture du courrier, rapatriement du courrier via POP et envoie du courrier via SMTP).

Reportez-vous à l'une des sections précédentes pour installer et configurer celui que vous préférez. Ensuite, quand vous atteindrez la section *Tester*, essayer d'envoyer un courrier électronique sur un compte local de votre passerelle de courrier.

8.4 Lire le courrier

Si votre programme ne fait pas tout lui-même, vous pouvez installer elm, pgp, mush, pine, etc. Beaucoup de bons programmes sont disponibles librement et gratuitement pour les plate-formes Linux.

8.5 Tester

Pour vérifier si votre serveur POP est bien installé tapez

```
$ telnet machine 110
```

Si cela marche, vous obtiendrez quelque chose du genre « OK Pop server (...) starting ». Tapez « quit » !

Pour installer un canal ssh chiffré, testez d'abord votre serveur de courrier en tapant

```
$ ssh machine date
```

Si vous obtenez la date, cela devrait être bon. Remarquez que ssh ne vous demandera pas de mot de passe, en conséquence, vous devrez créer un fichier `.shosts` sur le serveur de courrier, contenant le nom du client. Pour tester la redirection de port ssh (que wgpop utilise), tapez

```
$ ssh -n -f -L 12314:localhost:110 machine sleep 30
```

puis

```
$ telnet localhost 12314
```

Alors vous aurez bon espoir de voir la bannière de la poste centrale POP. Si vous n'utilisez pas ssh, n'oubliez pas de commenter $ssh dans le script gwpop. Pour vérifier si procmail tourne essayez `procmail~-v`.

8.6 Utilisation

Maintenant vous pouvez éditer le script Perl gwpop pour vérifier que tout va bien, puis lancez gwpop :

```
$ gwpop -v votre-nom-d'utilisateur
POP password on mailhost: votre-mot-de-passe-secret
```

Si les « messages d'erreurs » de gwpop vous parraissent normaux, le courrier sera téléchargé depuis votre poste principal et placé sur votre machine locale à l'endroit que vous aurez indiqué à gwpop (faîtes des tests avec quelques courriers électroniques).

Vous pouvez aussi utiliser gwpop en tant que démon :

```
$ gwpop -d $HOME/tmp votre-nom-d'utilisateur
```

Les messages de gwpop seront envoyés à syslog et gwpop tournera indéfiniment en tâche de fond ; un signal HUP forcera gwpop à aller chercher votre courrier.

Vous pouvez télécharger les logiciels pop utilisés ici sur :

- `ftp://ftp.unina.it/pub/Unix/pkgs/network/mail/gwpop`
- `ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail`
- `http://www.cs.hut.fi/ssh/`

9 Remerciements

Les personnes suivantes ont aidé à rassembler les informations et les expériences qui ont rendu possible l'écriture de ce document :

Steve Robbins, Ian Kluft, Rich Braun, Ian Jackson, Syd Weinstein, Ralf Sauther, Martin White, Matt Welsh, Ralph Sims, Phil Hughes, Scot Stevenson, Neil Parker, Stephane Bortzmayer et spécialement merci à Vince Skahan pour sa large contribution.

Eric S. Raymond a revu ce document, a corrigé différentes fautes et a intégré la section *How Electronic Mail Works* de son Mail User's HOWTO.

Hitoshi Hayakawa a vérifié la section sur qmail, Jun Morimoto a ajouté plusieurs remarques à propos de popclient et de fetchmail et Takeo Nakano qui à forcé ispell à vérifier l'orthographe de ce document :-)

Si j'ai oublié quelqu'un, toutes mes excuses : envoyez-moi un courrier électronique.