

# Internet Mail Access Protocol

IMAP est un protocole de relève des messages électroniques, fonctionnellement comparable à POP.

Il présente toutefois de nombreux avantages et l'on peut se demander pourquoi il n'est pas plus souvent utilisé.

Ce chapitre essaye de montrer les avantages de ce protocole et les limites des actuels clients de messagerie.

Pour y arriver, nous utiliserons comme d'habitude des outils de base :

- Telnet, le terminal à tout faire,
- Ethereal, le sniffeur habituel.
- Thunderbird, un client de messagerie issu du projet Mozilla. Il existe pour les plate formes Windows, Linux et Mac OS X. Vous le trouverez sur le site du projet Mozilla<sup>1</sup>.

Deux serveurs IMAP seront testés :

- l'un, utilisant le format "mailbox", monté sur une machine Debian "Woody", conjointement avec un SMTP EXIM.
- l'autre, utilisant le format "maildir", monté sur une Debian "testing", toujours avec Postfix.

---

<sup>1</sup> Projet Mozilla : <http://www.mozilla.org/projects/thunderbird/>

## Plan du chapitre

Présentation générale.....	4
Pourquoi IMAP4 ?.....	4
Puisqu'on a POP3.....	4
Mais avec IMAP4.....	4
Ce que vous devez pouvoir faire avec IMAP4 :.....	5
Alors pourquoi POP3 encore ?.....	5
Démonstration.....	6
Les configurations utilisées.....	6
UW-IMAP.....	6
Cyrus21.....	6
Les tests.....	6
Configuration du client (MUA).....	6
Création de répertoires.....	7
Réception d'un premier mail.....	8
Manipulations diverses.....	10
Premières conclusions.....	21
Le protocole IMAP.....	23
Définition du protocole.....	23
Mode opératoire.....	23
Les serveurs IMAP courants.....	23
UW-IMAPD.....	23
CYRUS.....	24
Les outils à (presque) tout faire.....	24
Les comptes pour faire les manips.....	24
Premier contact.....	24
Quelques commandes simples avec Telnet.....	25
Préliminaires.....	26
Ouverture d'une session IMAP.....	26
Arranger son intérieur.....	26
Lire un message.....	27
La commande LIST.....	28
La commande SELECT.....	29
La commande FETCH.....	30
La commande STORE.....	31
La commande EXPUNGE.....	34
Conclusions.....	35
Emploi des commandes IMAP.....	35
Le format MAILBOX.....	36
Deux serveurs IMAP4.....	37
Le cas simple et facile à installer.....	37
Pour Mandrake.....	37
Pour Debian :.....	37
Pour les deux :.....	38
Le cas compliqué et difficile à installer.....	38
Pourquoi tout ça ?.....	38

Matériel requis.....	38
Cyrus.....	39
Installation de Cyrus :.....	39
installation de sasl :.....	40
Vérifications.....	41
saslauthd.....	41
Cyrus.....	42
Qu'avons-nous fait ?.....	46
Que reste-t-il à faire ?.....	47
Tests.....	48
Juste un détail.....	50
Plus loin avec Cyrus.....	51
Les dossiers partagés.....	51
Création d'un dossier partagé.....	51
Conclusions.....	60

## Présentation générale

Initialement, IMAP représentait un acronyme de : "Interactive Message Access Protocol". Le nom a été modifié en "Internet Message Access Protocol" Pour tenir compte des derniers ajouts au protocole (actuellement en version 4 révision 1).

Il est actuellement défini par la RFC 2060<sup>2</sup>.

## Pourquoi IMAP4 ?

### Puisqu'on a POP3...

POP3 remplit tout à fait son rôle de relève de courrier, nous l'avons vu. Alors pourquoi changer ?

POP3 permet de travailler en modes "hors-ligne" et "déconnecté", autrement dit, il est possible :

- de rapatrier tous ses messages en local et de les effacer du serveur, ce qui permet d'interrompre la connexion et de gérer ses messages localement (mode "hors-ligne"),
- de faire la même chose, mais en rapatriant une copie locale des messages, laissant les messages "originaux" sur le serveur (mode "déconnecté").

Le mode "hors-ligne" est tout à fait utilisable si l'on ne gère sa messagerie que depuis un seul poste de travail, ce qui n'est pas toujours le cas.

Le mode "déconnecté" permet quant à lui une gestion depuis plusieurs postes, mais pose tout de même le problème de la purge du serveur. En effet, il faudra bien faire de la place de temps en temps si l'on ne veut pas voir sa boîte exploser. Et les messages une fois détruits sur le serveur ne pourront plus y être remis autrement qu'en se les renvoyant.

Lorsque l'on est dans des conditions de connexion difficiles, POP3 se révèle peu puissant pour se tirer d'embarras si un message volumineux se trouve dans la file. Il est possible, en exploitant toutes les finesses de POP3, d'éliminer ce message ou du moins de ne pas le rapatrier, mais peu de MUA savent gérer ces possibilités et le message non lu représentera toujours un écueil, à chaque consultation.

### Mais avec IMAP4...

Ici, le protocole autorise des manipulations infiniment plus souples. De plus, et c'est probablement là le point le plus décisif, les messages peuvent être entièrement gérés en restant sur le serveur. IMAP propose en effet les possibilités suivantes :

- lecture des objets des messages seulement (sans le corps),
- création de dossiers **sur le serveur**,
- déplacement de messages sur le serveur d'un dossier à l'autre, **sans avoir à rapatrier tout le message en local**,
- effacement de message sans l'avoir lu,

---

<sup>2</sup> RFC 2060 : <http://www.ietf.org/rfc/rfc2060.txt>

- lecture des messages en les laissant sur le serveur,
- marquage des messages sur le serveur...

Toutes ces possibilités nécessitent bien entendu d'être connecté en permanence, donc en mode interactif, d'où le nom initial du protocole.

Mais IMAP fait encore plus, dans la mesure où les modes "hors-ligne" et "déconnecté" sont également possibles.

### **Ce que vous devez pouvoir faire avec IMAP4 :**

- consulter seulement les objets des messages,
- effacer, déplacer des messages sans les lire, éventuellement avec des règles de tri automatiques,
- rapatrier en local certains messages et pas d'autres, en faisant une copie ou un déplacement, éventuellement avec des règles de tri automatiques,
- recopier sur le serveur des messages que vous avez en local,
- et d'autres choses encore.

Vous le voyez, il semble n'y avoir aucune bonne raison de ne pas passer à IMAP.

### **Alors pourquoi POP3 encore ?**

S'il ne semble y avoir que de bonnes raisons de passer à IMAP, il y en a aussi (mais sont-elles bonnes ?) pour rester sur POP3.

IMAP4 est un protocole beaucoup plus compliqué que POP3 et pour cause, il est plus puissant. Cette complexité relative amène plusieurs effets négatifs :

- tous les fournisseurs de services Internet ne proposent pas encore de serveur IMAP, et ceux qui en proposent, pour des raisons diverses, les amputent parfois de certaines de leurs possibilités,
- rares sont les clients de messagerie (MUA) qui gèrent toutes les possibilités offertes par IMAP4, si l'on se limite à ce que sait faire POP3, alors, autant utiliser POP3,
- garder tous ses messages sur le serveur, même bien classés dans divers dossiers n'a pas que des avantages, l'espace disponible est souvent limité (5 Mo, parfois moins, très rarement plus) et le stockage sur le serveur va rapidement remplir cet espace. Il faudra donc adopter des stratégies de purge qui restreindront les avantages du système.

Mais nous sommes ici pour parler d'IMAP4. Pas moins de 25 commandes alors que POP3 n'en propose que 12. Nous ne les verrons pas toutes en détail, le but étant d'avantage de comprendre l'intérêt du protocole que de le manipuler avec telnet :-)

## Démonstration

### Les configurations utilisées

Nous disposons de deux serveurs IMAP différents. Pour l'instant, nous nous contenterons de les utiliser, nous verrons plus en détails comment les installer plus tard.

#### UW-IMAP

Un serveur développé à l'université de Washington, installé sur une Debian Woody. Le serveur SMTP utilisé est EXIM, installé par défaut par Debian. C'est un bon SMTP, plus souple que Postfix, mais aussi plus délicat à configurer. La machine s'appelle poétiquement gw2.maison.mrs.

Ce serveur utilise le format "MAILBOX".

#### Cyrus21

Un serveur développé à l'université de Carnegie Mellon, installé sur une Debian "testing". Le SMTP employé ici est Postfix. C'est un bon SMTP, moins souple qu'EXIM mais plus facile à configurer :). La machine s'appelle mythologiquement cyclope.maison.mrs.

Ce serveur utilise le format "MAILDIR".

### Les tests

#### Configuration du client (MUA)

Sur chacune de ces machines, un compte de messagerie est créé :

- [testimap@gw2.maison.mrs](mailto:testimap@gw2.maison.mrs) pour le serveur UW-imap,
- [testimap@cyclope.maison.mrs](mailto:testimap@cyclope.maison.mrs) pour le serveur Cyrus.

Un client de messagerie, "Thunderbird", est installé sur une troisième machine : pchris2.maison.mrs, qui fonctionne sous Mandrake 9.1 (nous avons les moyens pour travailler correctement).

Thunderbird est donc issu du projet Mozilla. C'est le client de messagerie qui a été extrait et très légèrement modifié.

Si vous utilisez la suite Mozilla, ce qui est tout à fait recommandable, Thunderbird ne sera donc pas nécessaire. L'avantage de Thunderbird est qu'il ne nécessite pas d'installation. Vous téléchargez le pack, vous le désarchivez où vous voulez, vous donnez les bons droits d'exécution et c'est tout. Aussi bien sous Linux, quelle qu'en soit la distribution, que sous Windows.

Le tout monté en réseau local, bien entendu.

Nous allons créer ces deux comptes sur Thunderbird :



Deux remarques immédiates :

- les deux comptes contiennent déjà deux dossiers, "Inbox" et "Trash",
- sur gw2 (uw-imap) les deux dossiers sont au même niveau de hiérarchie, alors que sur cyclope (cyrus), "Trash" est un sous dossier de "Inbox". Cette subtilité trouvera son explication plus loin dans cet exposé.

## Création de répertoires

Ceci n'est pas un cours sur l'emploi de Thunderbird. Nous nous dispenserons donc de développer le mode opératoire.

Thunderbird aime bien disposer de répertoires supplémentaires :

- Sent, pour stocker les messages envoyés,
- Drafts, pour stocker les brouillons,
- Templates, pour stocker les modèles.

Nous allons les créer pour chaque compte, sur les serveurs respectifs.



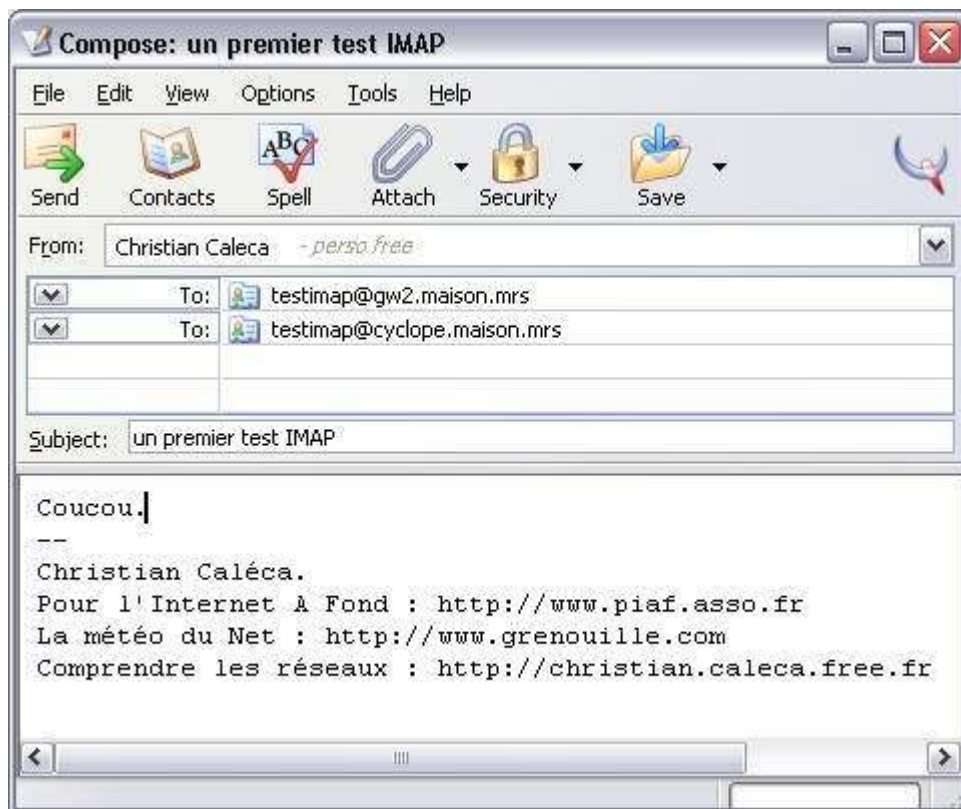
Encore deux remarques :

- avec uw-imapd, il n'est pas possible de créer ces dossiers dans "Inbox", tous les dossiers sont obligatoirement au même niveau de hiérarchie,
- avec Cyrus, les dossiers ne peuvent être créés que dans "Inbox" (ou dans un sous dossier de "inbox"). Il est possible de construire une arborescence complexe, même si ce n'est pas forcément souhaitable.

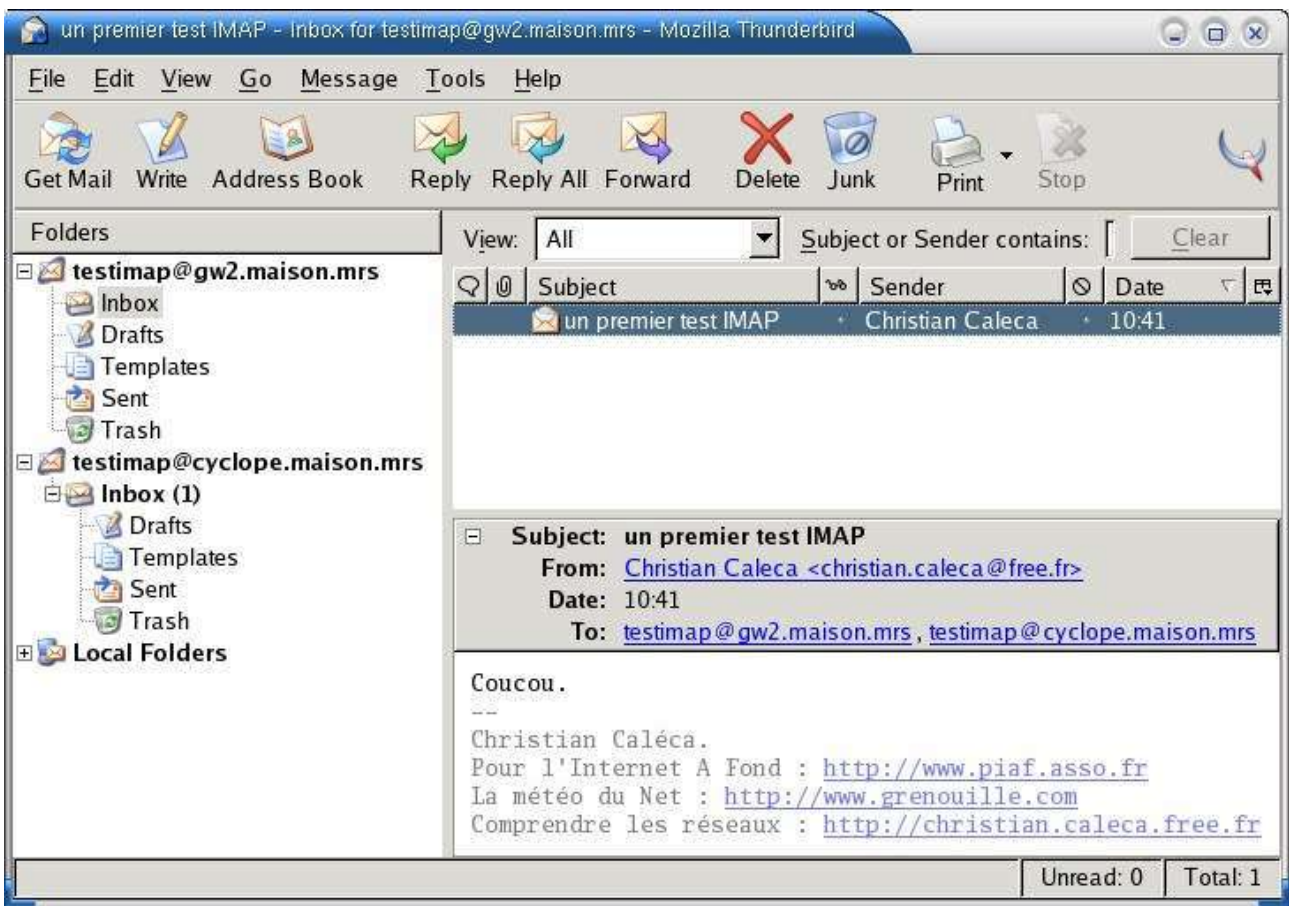
## Réception d'un premier mail

Nous avons plus de moyens que vous ne pensez. Depuis une quatrième machine (Windows, celle là, mais qui utilise aussi Thunderbird), nous envoyons un message sur ces deux comptes :



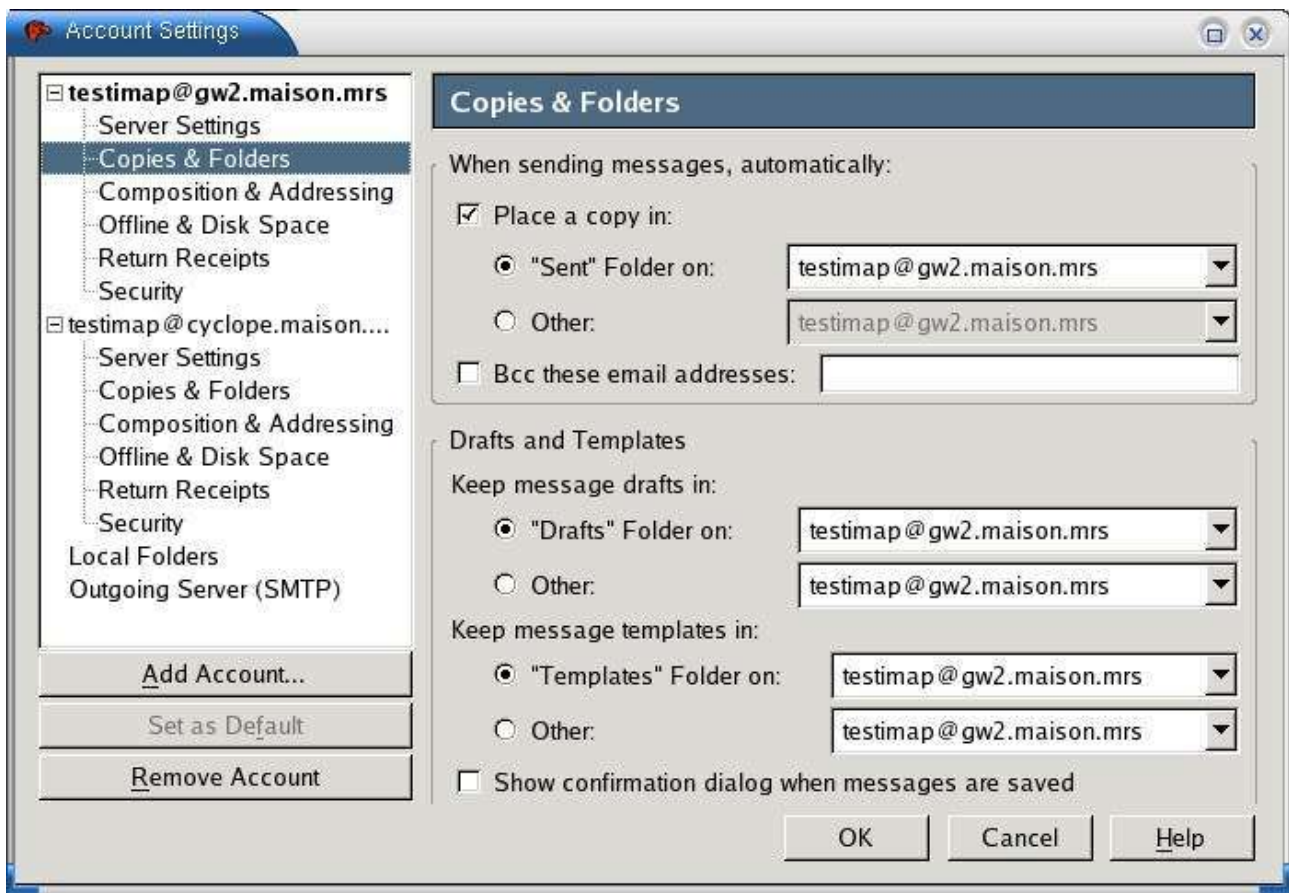


Bien entendu, ça fonctionne et nous retrouvons sur pchris2 ce message dans chaque BAL :

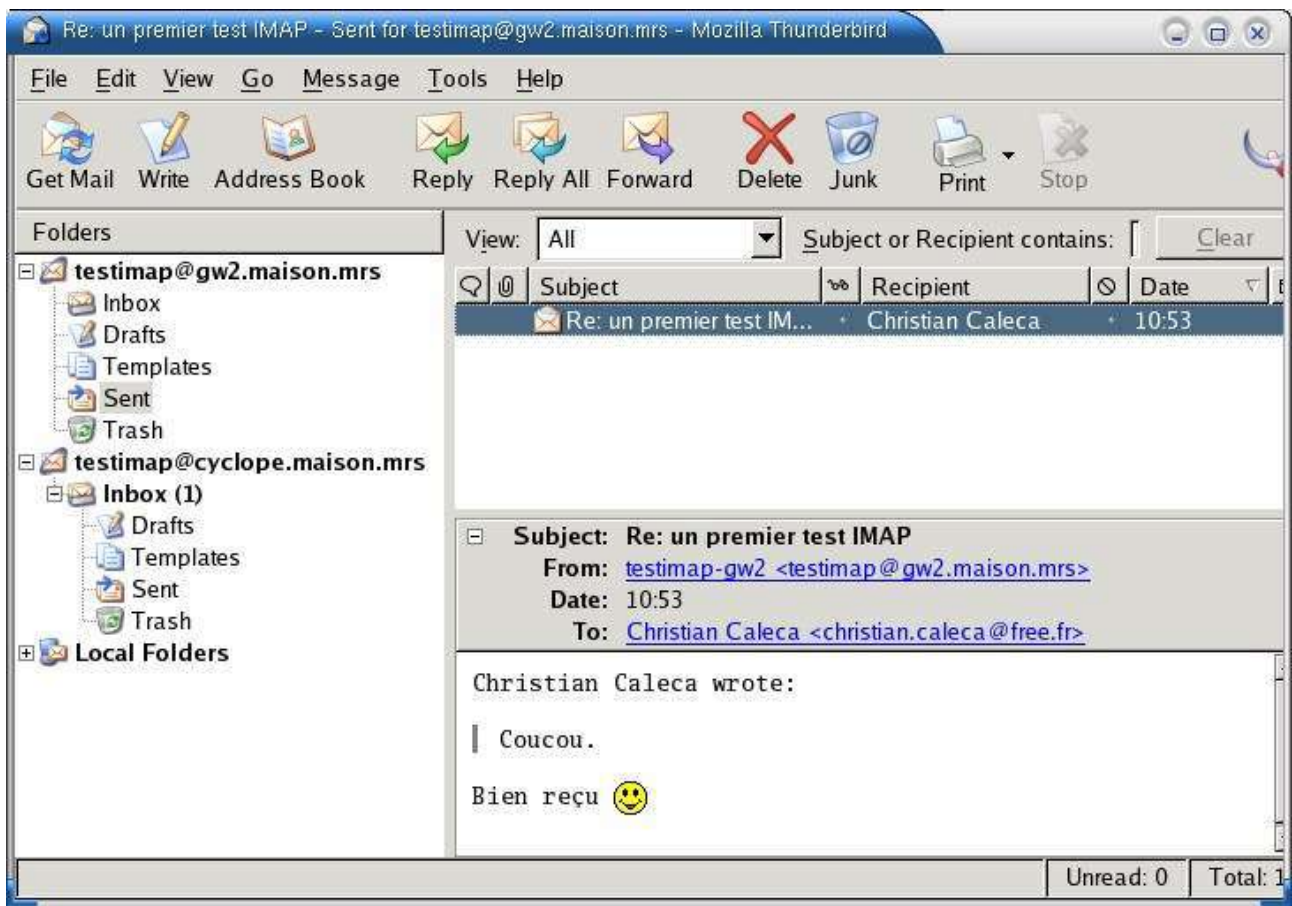


## Manipulations diverses

Voyons un peu la configuration de notre Thunderbird :



Par défaut, Thunderbird place une copie des messages envoyés dans le dossier "Sent", sur le serveur IMAP du compte employé. Vérifions ça en répondant à ce premier message depuis le compte sur gw2 :



Il y est.

Le dossier "Sent" est bien sur le serveur IMAP de gw2.maison.mrs et le message envoyé s'y trouve bien. Nous allons le vérifier tout de suite, puisque nous avons le serveur sous la main :

```
/home/testimap# ls -la
total 28
drwxr-xr-x  2 testimap nogroup   4096 Dec 20 10:53 .
drwxrwsr-x  6 root      staff    4096 Nov 30 16:42 ..
-rw-r--r--  1 testimap nogroup    28 Dec 20 10:20 .mailboxlist
-rw-----  1 testimap nogroup   513 Dec 20 10:19 Drafts
-rw-----  1 testimap nogroup  1190 Dec 20 10:53 Sent
-rw-----  1 testimap nogroup   513 Dec 20 10:20 Templates
-rw-----  1 testimap nogroup   513 Dec 20 10:03 Trash
```

Nous avons bien quatre fichiers qui correspondent aux quatre dossiers créés et un cinquième, caché, qui s'appelle .mailboxlist. Etant d'un naturel curieux, impossible de résister à l'envie de regarder son contenu :

```
/home/testimap# cat .mailboxlist
Trash
Drafts
Sent
Templates
```

Un peu décevant... Il ne contient que la liste des noms des répertoires.

Voyons maintenant le contenu du fichier "Sent" :

```
sysop:/home/testimap# cat Sent
From MAILER-DAEMON Sat Dec 20 10:20:02 2003
```

```
Date: 20 Dec 2003 10:20:02 +0100
From: Mail System Internal Data <MAILER-DAEMON@sysop.eme-enseignement.fr>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
X-IMAP: 1071912002 0000000000
Status: RO
```

This text is part of the internal format of your mail folder, and is not a real message. It is created automatically by the mail system software. If deleted, important folder data will be lost, and it will be re-created with the data reset to initial values.

```
From testimap@sysop.eme-enseignement.fr Sat Dec 20 10:53:12 2003 +0100
Status: R
X-Status:
X-Keywords:
Message-ID: <3FE41C07.1030504@gw2.maison.mrs>
Date: Sat, 20 Dec 2003 10:53:11 +0100
From: testimap-gw2 <testimap@gw2.maison.mrs>
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6b) Gecko/20031205 Thunderbird/0.4
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: Christian Caleca <christian.caleca@free.fr>
Subject: Re: un premier test IMAP
References: <3FE4192C.7040107@free.fr>
In-Reply-To: <3FE4192C.7040107@free.fr>
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 8bit
```

Christian Caleca wrote:

> Coucou.

Bien reçu :)

Intéressons nous pour l'instant à ce qui est surligné : c'est bien le texte de la réponse faite.

Il n'a bien entendu pas échappé à votre sagacité que le dossier "Inbox" n'est pas ici. C'est tout simplement qu'il est ailleurs. Il est dans le spool de messagerie, directement alimenté par le SMTP, via l'agent de distribution local (MDA). Avec EXIM vous le trouverez dans /var/spool/mail

```
sysop:/var/spool/mail# ls
chris testimap
```

Profitons-en pour voir ce qu'il y a dedans :

```
sysop:/var/spool/mail# cat testimap
From christian.caleca@free.fr Sat Dec 20 10:40:53 2003
Return-path: <christian.caleca@free.fr>
Envelope-to: testimap@gw2.maison.mrs
Received: from pchris.maison.mrs ([192.168.0.10] helo=free.fr)
        by sysop.eme-enseignement.fr with esmtp (Exim 3.35 #1 (Debian))
        id 1AXdbQ-0000u2-00; Sat, 20 Dec 2003 10:40:52 +0100
Message-ID: <3FE4192C.7040107@free.fr>
Date: Sat, 20 Dec 2003 10:41:00 +0100
From: Christian Caleca <christian.caleca@free.fr>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6b) Gecko/20031205 Thunderbird/0.4
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: testimap@gw2.maison.mrs, testimap@cyclope.maison.mrs
Subject: un premier test IMAP
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 8bit
X-IMAPbase: 1071913280 2
Status: RO
X-Status: DA
X-Keywords:
X-UID: 1

Coucou.
--
```

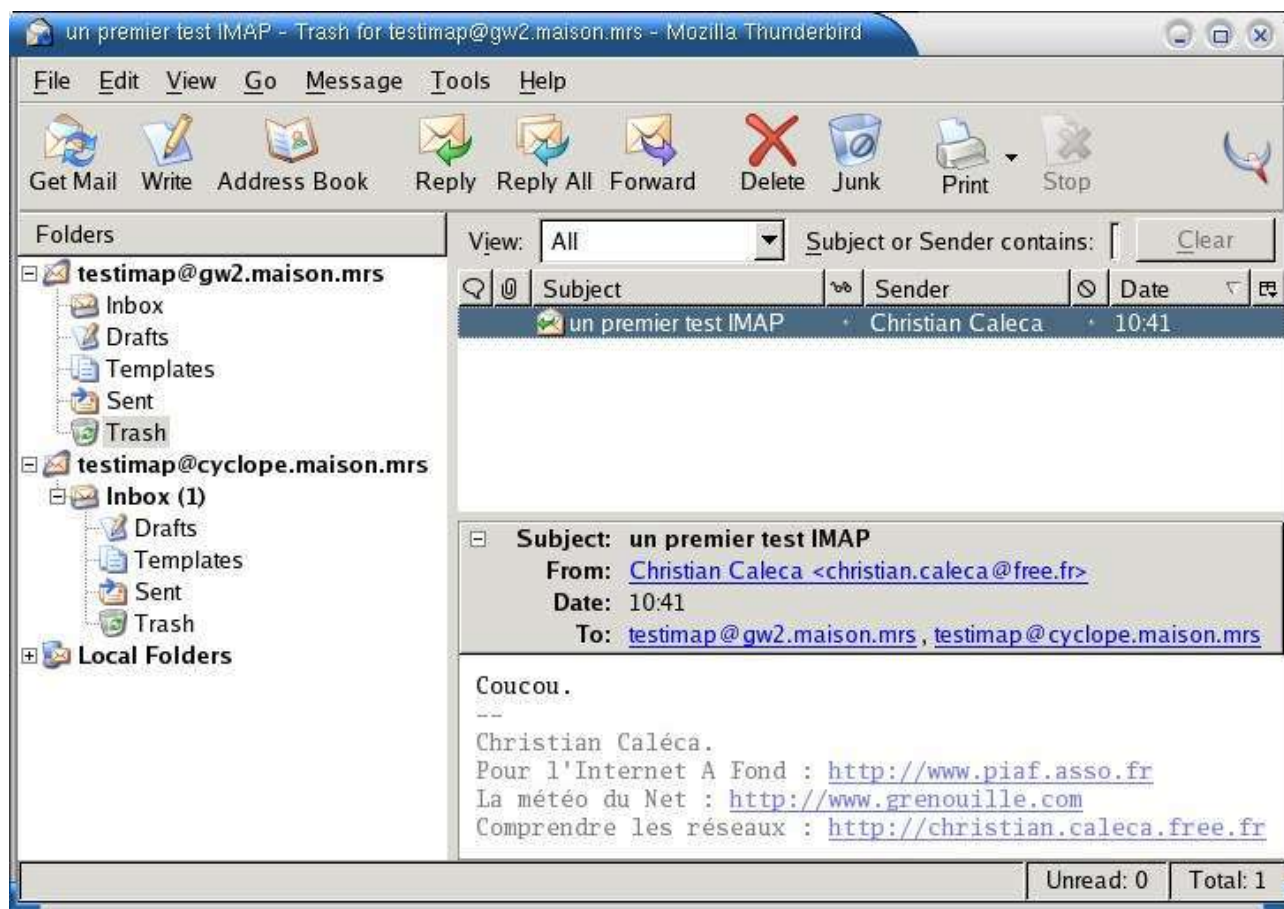


Christian Caléca.  
 Pour l'Internet A Fond : <http://www.piaf.asso.fr>  
 La météo du Net : <http://www.grenouille.com>  
 Comprendre les réseaux : <http://christian.caleca.free.fr>

Le message que l'on a reçu. C'est réconfortant.

## Suppression d'un message dans "Inbox"

Le message initial n'ayant pas d'intérêt, nous allons le détruire. Nous devrions théoriquement le retrouver dans la poubelle (Trash) :



Tout va bien, tout se passe comme prévu.

Comme ce message n'a toujours pas d'intérêt, même dans la poubelle, nous vidons aussi la poubelle.

Bien. Vous êtes bien assis ? Alors, allons vérifier tout ça sur le serveur :

```
/home/testimap# cat Trash
From MAILER-DAEMON Sat Dec 20 11:07:13 2003
Date: 20 Dec 2003 11:07:13 +0100
From: Mail System Internal Data <MAILER-DAEMON@sysop.eme-enseignement.fr>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
Message-ID: <1071914833@sysop.eme-enseignement.fr>
X-IMAP: 1071911016 0000000001
Status: RO

This text is part of the internal format of your mail folder, and is not
a real message. It is created automatically by the mail system software.
If deleted, important folder data will be lost, and it will be re-created
with the data reset to initial values.
```

```
From christian.caleca@free.fr Sat Dec 20 10:40:53 2003
Return-path: <christian.caleca@free.fr>
Envelope-to: testimap@gw2.maison.mrs
Received: from pchris.maison.mrs ([192.168.0.10] helo=free.fr)
        by sysop.eme-enseignement.fr with esmtp (Exim 3.35 #1 (Debian))
        id 1AXdbQ-0000u2-00; Sat, 20 Dec 2003 10:40:52 +0100
Message-ID: <3FE4192C.7040107@free.fr>
Date: Sat, 20 Dec 2003 10:41:00 +0100
From: Christian Caleca <christian.caleca@free.fr>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6b) Gecko/20031205 Thunderbird/0.4
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: testimap@gw2.maison.mrs, testimap@cyclope.maison.mrs
Subject: un premier test IMAP
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 8bit
Status: RO
X-Status: A
X-Keywords:

Coucou.
--
Christian Caléca.
Pour l'Internet A Fond : http://www.piaf.asso.fr
La météo du Net : http://www.grenouille.com
Comprendre les réseaux : http://christian.caleca.free.fr
```

Ca c'est c.. ennuyeux. Bien qu'effacé, le message y est toujours !

Et dans le spool, le premier message reçu, puis effacé, y est-il toujours lui aussi ?

```
sysop:/var/spool/mail# cat testimap
From christian.caleca@free.fr Sat Dec 20 10:40:53 2003
Return-path: <christian.caleca@free.fr>
Envelope-to: testimap@gw2.maison.mrs
Received: from pchris.maison.mrs ([192.168.0.10] helo=free.fr)
        by sysop.eme-enseignement.fr with esmtp (Exim 3.35 #1 (Debian))
        id 1AXdbQ-0000u2-00; Sat, 20 Dec 2003 10:40:52 +0100
Message-ID: <3FE4192C.7040107@free.fr>
Date: Sat, 20 Dec 2003 10:41:00 +0100
From: Christian Caleca <christian.caleca@free.fr>
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6b) Gecko/20031205 Thunderbird/0.4
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: testimap@gw2.maison.mrs, testimap@cyclope.maison.mrs
Subject: un premier test IMAP
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 8bit
X-IMAPbase: 1071913280 2
Status: RO
X-Status: DA
X-Keywords:
X-UID: 1

Coucou.
--
Christian Caléca.
Pour l'Internet A Fond : http://www.piaf.asso.fr
La météo du Net : http://www.grenouille.com
Comprendre les réseaux : http://christian.caleca.free.fr
```

Oui...

Ça voudrait dire que petit à petit, l'espace alloué va s'encombrer de déchets et au final, la BAL va exploser alors même qu'elle sera considérée comme vide ?

La réponse est oui, si l'on ne prend pas une précaution supplémentaire : le compactage des dossiers.

En cliquant sur Inbox du bouton droit et en faisant "Compact This Folder" et en répétant la même opération sur "Trash", nous allons remédier au problème :

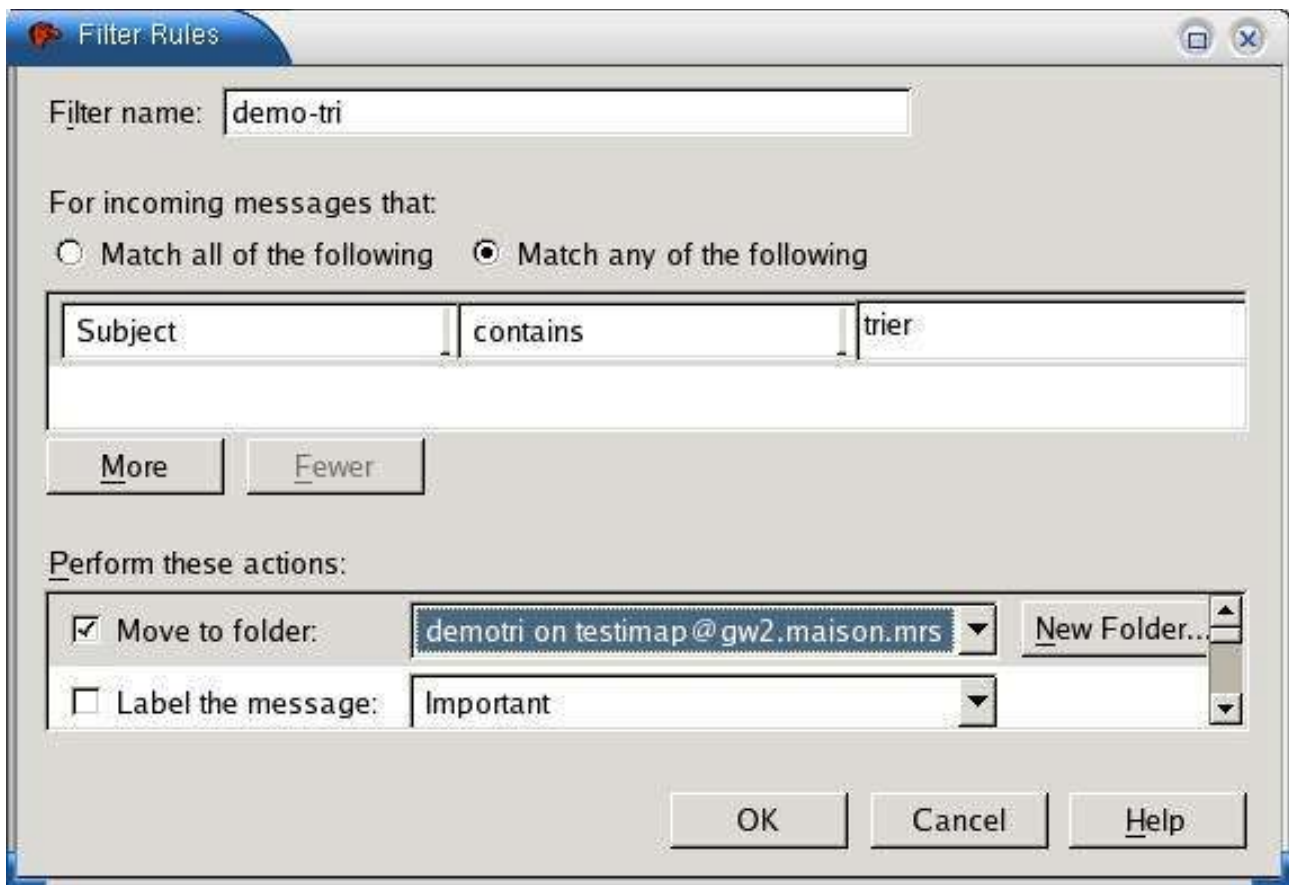
```
/var/spool/mail# ls -l
total 48
-rw-rw----  1 testimap mail          0 Dec 20 11:33 testimap
```

Le fichier existe toujours, mais fait 0 octets, ce qui prouve qu'il est vide.

Il est donc primordial, avec IMAP, de penser à compacter régulièrement les dossiers de la messagerie.

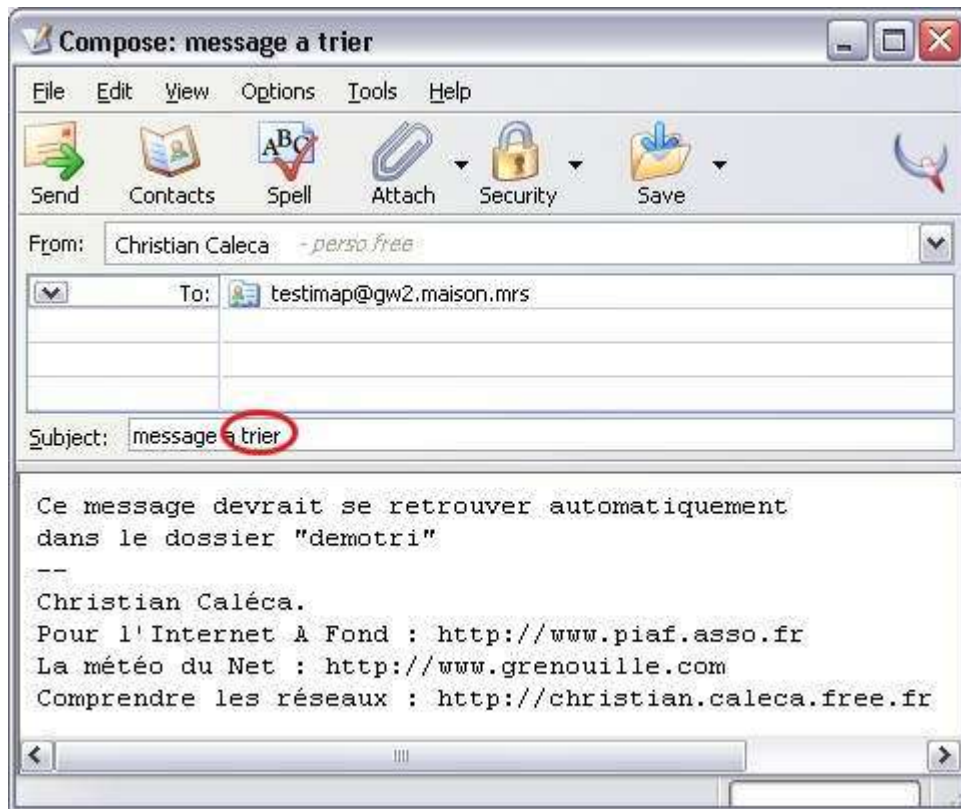
## Déplacement de messages

Nous allons créer pour le compte sur gw2.maison.mrs une règle de filtrage qui va déplacer tout message contenant le mot "trier" dans un dossier spécial intitulé "demotri" et créé à cet effet.

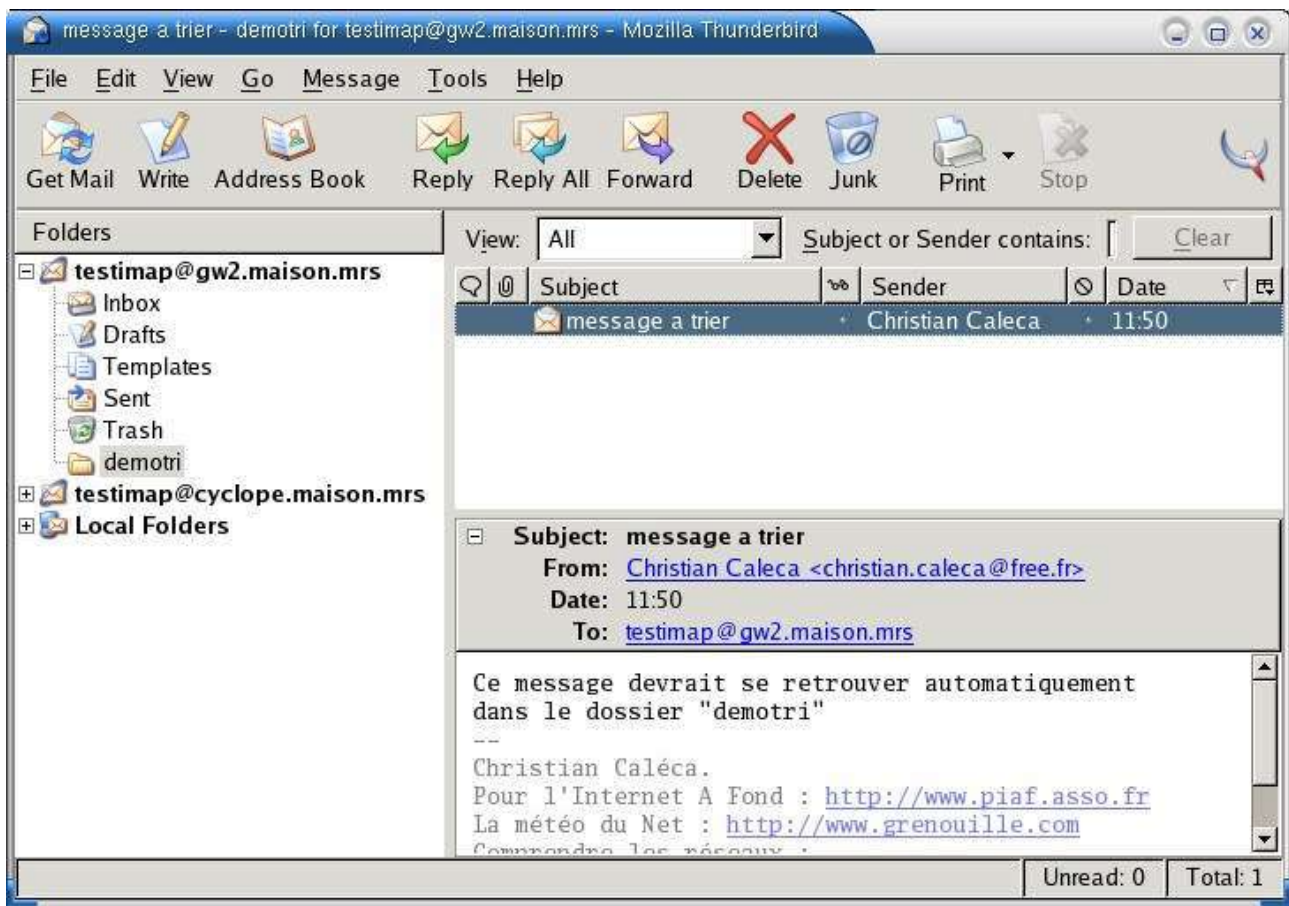


et nous envoyons un message :





Et chez le destinataire :



Ça fonctionne. Le seul fait de lire sa messagerie va faire que le message sera déplacé dans le dossier "demotri" sans qu'il ait été au préalable rapatrié chez le client.

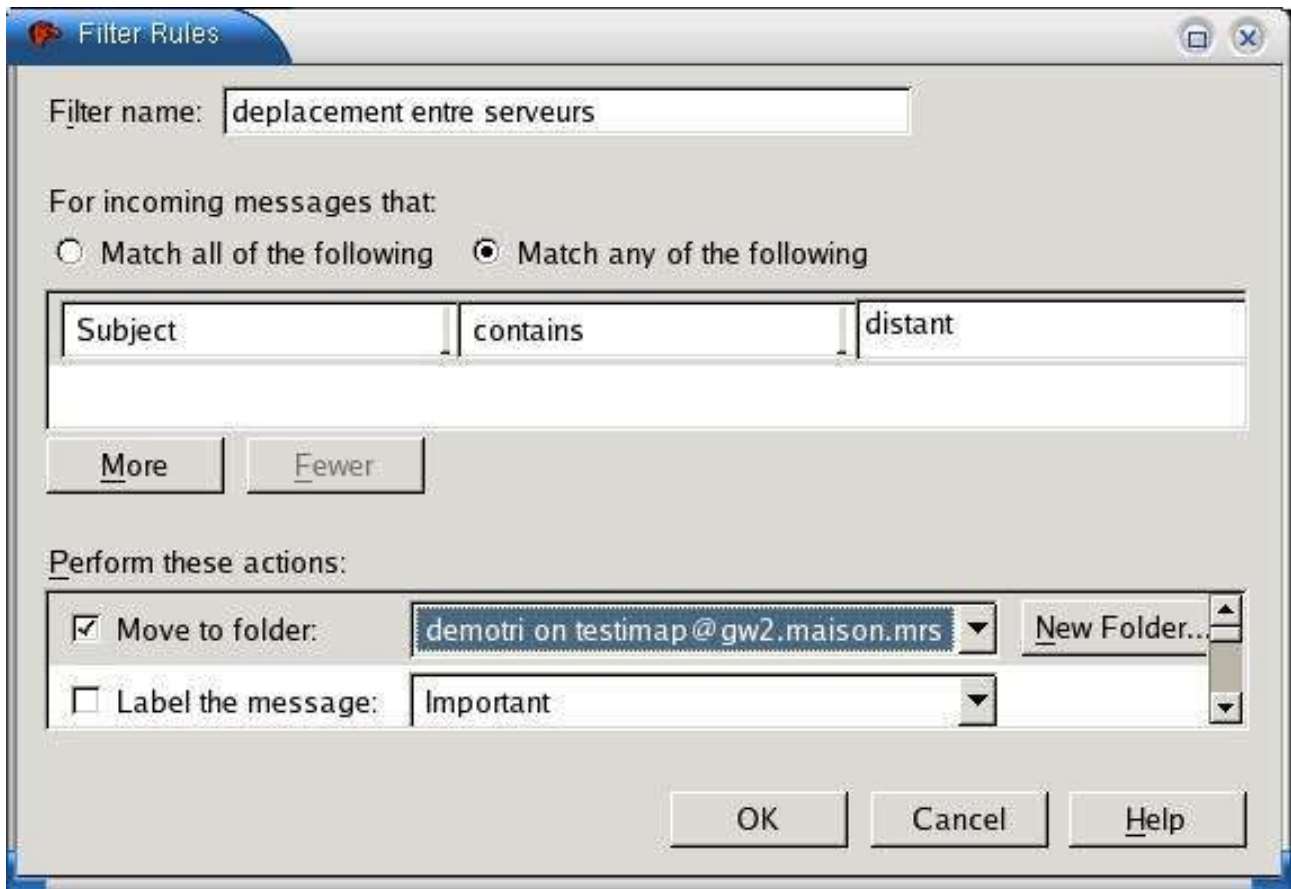
On efface ce message sans intérêt, on vide la poubelle et au bout du compte, notre BAL contiendra toujours trois exemplaires de ce message, invisibles, mais bien présents :

- Dans Inbox, parce que le déplacement n'est en réalité qu'une copie suivie d'un effacement,
- dans demotri,
- dans Trash.

Pensez donc à compacter les dossiers souvent ;-)

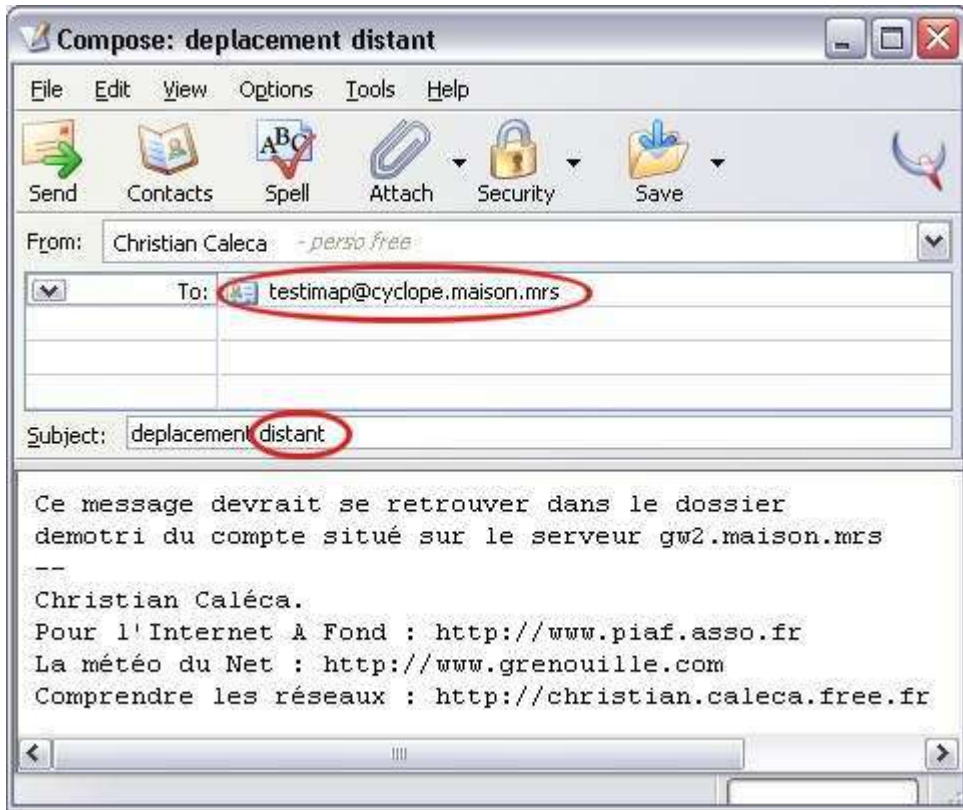
Plus fort encore, nous allons créer une règle de tri qui fera que, lorsqu'un message à destination de [testimap@cyclope.maison.mrs](mailto:testimap@cyclope.maison.mrs) contient le mot "distant" dans son objet, il faudra le déplacer dans le répertoire "demotri" du compte [testimap@gw2.maison.mrs](mailto:testimap@gw2.maison.mrs).

Autrement dit, nous allons déplacer un message d'un serveur à l'autre.

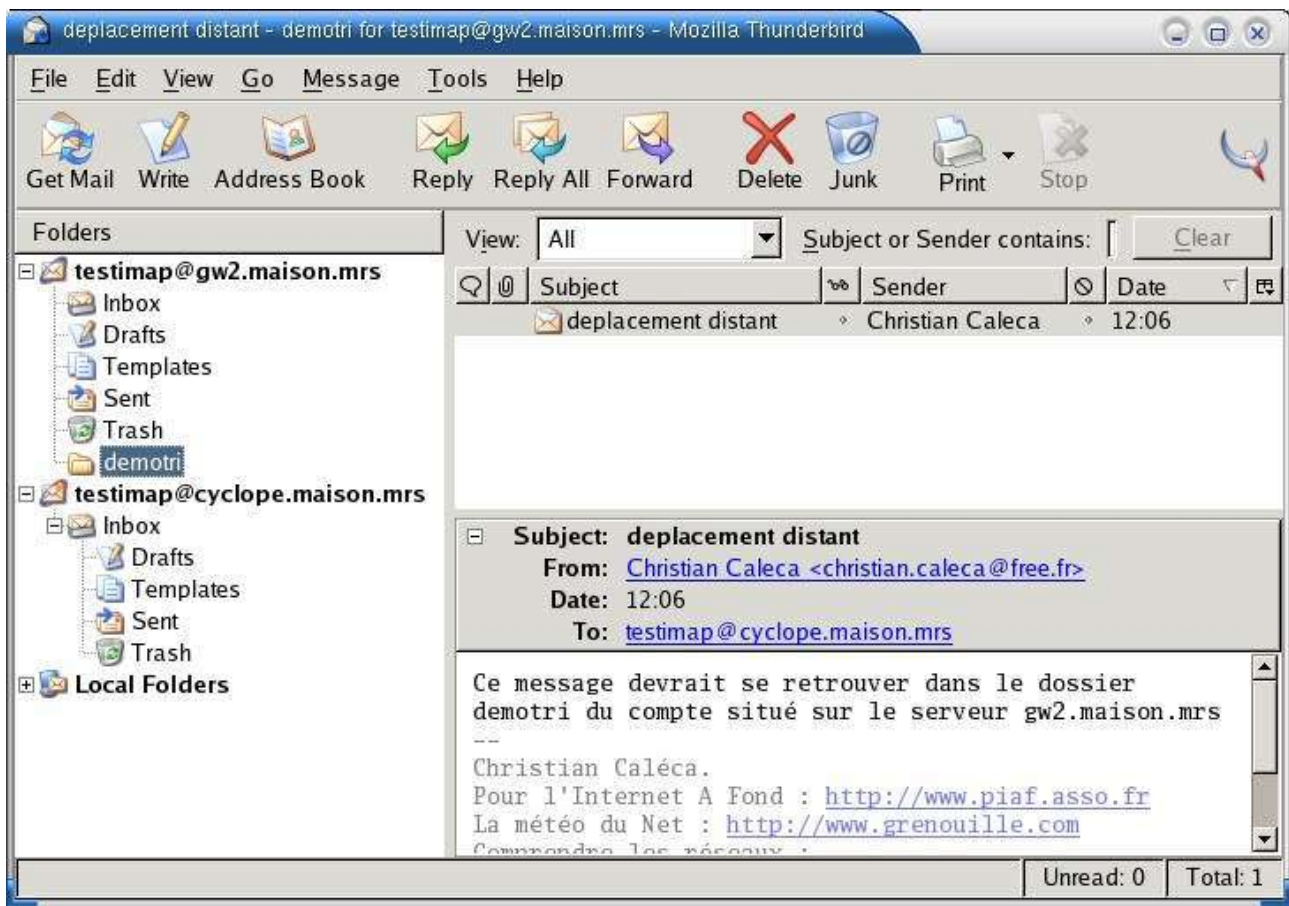


Répetons-le, cette règle est écrite pour le compte testimap@cyclope.maison.mrs !

Envoi du message :



Et réception :



### Le cas d'un gros message encombrant.

Vous êtes perdu quelque part de l'autre côté de la fracture numérique et ne disposez que d'une méchante connexion RTC qui plafonne à 28800 bps et qui se déconnecte toutes les cinq minutes, à cause de la mauvaise qualité de votre ligne téléphonique. Je peux vous indiquer des endroits en France où c'est comme ça que ça se passe.

Comme dans ce cas, vous avez pris la précaution de faire afficher la taille des messages, vous constatez que celui-ci fait 805 Ko, qu'avec votre connexion pourrie, vous n'arriverez jamais à le télécharger, IMAP vous sauve.

En effet, à ce stade, le message n'est pas téléchargé en local. Aussi longtemps que vous ne cliquerez pas dessus du bouton gauche, il ne se téléchargera pas.

Cliquez donc dessus **du bouton droit**, demandez de le déplacer dans le dossier "Lire\_plus\_tard", que vous avez créé à cet effet. Le déplacement aura lieu sans que le message ne soit téléchargé localement. Vous pourrez alors aller le lire plus tard, lorsque vous aurez retrouvé une connexion de bonne qualité.

## Premières conclusions

Si cette démonstration ne vous a pas convaincu de l'intérêt d'IMAP, c'est que vous n'avez pas besoin de consulter votre messagerie depuis des machines différentes, que vous êtes suffisamment sûr de la

fiabilité de votre machine locale pour ne pas souhaiter conserver vos messages importants sur le serveur de votre fournisseur, que vous n'avez jamais été confronté au blocage de votre messagerie parce que vous avez une connexion tellement minable qu'un gros message ne peut jamais être rapatrié à cause des déconnexions.

IMAP propose beaucoup de fonctionnalités, c'est une autre affaire que d'en disposer avec son MUA. Thunderbird gère bien mieux l'IMAP que ne le fait Outlook Express, par exemple, qui ne sait pas appliquer de règles de filtrage sur les dossiers IMAP. Cependant, il n'est pas parfait non plus. Il n'est pas possible par exemple de définir simplement une règle de tri en fonction de la taille des messages.

Pourquoi avons-nous fait ces manipulations surtout avec UW-imap ? Parce que c'est le serveur dont la structure est la plus simple. Mais rassurez-vous, nous verrons Cyrus plus en détails dans la suite de cet exposé.

De ce que nous avons vu pour l'instant, retenons que Cyrus offre plus de souplesse dans l'organisation des répertoires que ne le fait UW-imap. Nous verrons plus loin que ce n'est pas son seul avantage. Mais en ce qui concerne le protocole IMAP lui-même, les deux serveurs se comportent de la même manière.



# Le protocole IMAP

Comme nous l'avons dit plus haut, tous les clients de messagerie ne gèrent pas IMAP au mieux. Voir ce que l'on peut faire avec ce protocole à travers un MUA ne donnera pas un aperçu de toutes les possibilités de ce protocole. Et puis, mettre un peu les mains dans le cambouis, ça ne fait pas de mal.

## Définition du protocole

RTFRFC 2060, comme on dit chez les branchés.

Il en existe une traduction officieuse en français<sup>3</sup>. Elle est suffisamment bien faite pour ne pas manquer de l'utiliser. Il n'est donc pas question ici de reprendre tout ce qui est dit dedans. Nous allons plutôt essayer de vérifier quelques points par la pratique.

Il est tout de même peut-être bon de rappeler que IMAP est un protocole d'application, qu'il s'appuie sur TCP et que le serveur écoute par défaut sur le port 143.

## Mode opératoire

Il n'est pas conseillé de "bricoler" avec des serveurs IMAP de production, par exemple celui (ceux) de votre FAI. Généralement, ces serveurs sont très sollicités. Avec le développement sans cesse croissant des connexions permanentes à haut débit, les habitudes changent.

- Les Boîtes aux lettres, consultées autrefois une ou deux fois par jour le sont maintenant plusieurs fois par heure,
- la taille des messages, grâce aux hauts débits et à cause de mauvaises habitudes a une forte tendance à augmenter.

Tout ceci fait que les serveurs SMTP/POP/IMAP voient leur charge augmenter dans d'énormes proportions.

Pour au moins ces raisons, il est bien plus convenable de bidouiller sur un serveur IMAP "maison". Si l'on dispose d'une (voire plusieurs) machine(s) sous Linux, ce n'est pas obligatoirement compliqué à mettre en oeuvre.

De plus, en ayant complètement la main sur les machines hébergeant les serveurs, nous pourrions modifier les configurations et voir de près comment les boîtes sont construites.

## Les serveurs IMAP courants

Nous en avons déjà parlé, rappelons-le :

### UW-IMAPD

UW IMAPD est développé par l'université de Washington. Il s'appuie sur le format "mailbox" pour stocker les messages. C'est un format classique que les MDA savent généralement bien gérer. En

---

<sup>3</sup> FRC 2060 en français : <http://jlr31130.free.fr/rfc2060.html>

gros, les messages sont enregistrés dans un fichier unique, pour un utilisateur donné. Le plus souvent, chaque utilisateur devra disposer d'un compte sur le serveur, même si ce compte ne lui permet pas d'ouvrir une session shell. Les dossiers IMAP que le client peut créer se situent au même niveau que INBOX et chaque dossier créé correspond sur le serveur à un fichier.

Ce genre de serveur est extrêmement facile à installer. Nous en utiliserons un exemple sur une Debian woody stable, avec EXIM comme MTA.

## CYRUS

Cyrus est " l'autre " serveur IMAP. Développé par l'université de Carnegie Mellon, il s'appuie sur le format "maildir". Les messages sont stockés chacun dans un fichier séparé, rangés dans un répertoire par utilisateur.

Les utilisateurs peuvent être enregistrés dans une base de données, un annuaire LDAP et n'ont absolument pas besoin de disposer d'un compte UNIX sur la machine serveur.

Ce serveur est plus souple, plus sûr, offre plus de possibilités et, bien entendu, est beaucoup plus difficile à installer et à configurer. Nous en verrons un exemple sur cyclope.maison.mrs, une machine Debian "testing".

D'autres serveurs existent, comme courier-imap, souvent utilisé avec le SMTP QMAIL. Il est plus proche de Cyrus que de UW-imap et utilise lui aussi le format "Maildir".

## Les outils à (presque) tout faire

Pour vérifier et expérimenter le protocole, en plus d'un MUA gérant proprement IMAP, nous utiliserons deux outils de base :

- Le sniffeur Ethereal pour regarder ce qu'il se passe sur le réseau,
- Telnet, pour manipuler les commandes du protocole au plus bas niveau.

## Les comptes pour faire les manips

Nous disposons de trois comptes sur trois serveurs différents (nous avons les moyens) :

**testimap@gw2.maison.mrs**      serveur uw-imap "packagé" dans la Debian Woody  
**testimap@cyclope.maison.mrs**    serveur Cyrus 21 de la Debian "testing"

Et d'un client sur un poste Windows XP (pour changer un peu de système).

## Premier contact

Comme je vous l'ai dit plus haut, une très honnête traduction de RFC 2060 existe, vous n'avez donc pas manqué de la lire , moi non plus d'ailleurs. Pas encore ? C'est pourtant nécessaire<sup>4</sup>... Nous savons donc déjà certaines choses :

- les commandes du client doivent commencer par un "tag", constitué d'une chaîne de caractères courte, différent à chaque commande,
- il existe trois états :

<sup>4</sup> RFC 2060 en français : <http://jlr31130.free.fr/rfc2060.html>



- non authentifié,
- authentifié,
- sélectionné.

Dans chacun de ces états, on a droit à un nombre plus ou moins grand de commandes. Dans le premier, non authentifié, on ne peut pas faire grand chose, mais tout de même...

Commençons doucement. Nous allons juste ouvrir une session sur chaque serveur et poser la question "CAPABILITY" à chacun d'eux.

C'est parti :

```
c:\> telnet gw2.maison.mrs 143
* OK [CAPABILITY IMAP4REV1 X-NETSCAPE LOGIN-REFERRALS AUTH=LOGIN] gw2.maison.mrs
IMAP4rev1 2001.315 at Thu, 6 Nov 2003 15:05:29 +0100 (CET)

A0001 CAPABILITY
* CAPABILITY IMAP4REV1 X-NETSCAPE NAMESPACE MAILBOX-REFERRALS SCAN SORT
THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND LOGIN-REFERRALS AUTH=LOGIN
A0001 OK CAPABILITY completed

A0002 LOGOUT
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
A0002 OK LOGOUT completed

c:\>telnet cyclope.maison.mrs 143
* OK cyclope.maison.mrs Cyrus IMAP4 v2.1.14-IPv6-Debian-2.1.14-1 server ready

A0001 CAPABILITY
* CAPABILITY IMAP4 IMAP4rev1 ACL QUOTA LITERAL+ MAILBOX-REFERRALS NAMESPACE UIDPLUS
ID NO ATOMIC RENAME UNSELECT CHILDREN MULTIAPPEND SORT THREAD=ORDEREDSUBJECT
THREAD=REFERENCES IDLE LISTEXT LIST-SUBSCRIBED ANNOTATEMORE
A0001 OK Completed

A0002 LOGOUT
* BYE LOGOUT received
A0002 OK Completed
```

Sans vraiment comprendre toutes les subtilités de "CAPABILITY", on voit clairement que Cyrus propose d'avantage de choses que UW-imap.

Notez que ce qui est important aux yeux de la norme, c'est que IMAP4rev1 figure dans la liste, ce qui est le cas des deux serveurs.

## Quelques commandes simples avec Telnet

Puisque maintenant, vous avez forcément lu les RFC 2060<sup>5</sup>, vous avez pu constater que IMAP4 est bien plus riche que POP3. Nous n'allons pas passer des pages et des pages à analyser toutes les commandes possibles, ce travail n'aurait d'ailleurs d'intérêt que pour ceux qui souhaitent développer un client (ou un serveur) IMAP. L'objectif de ce qui suit est juste de comprendre le principe de fonctionnement.

Les (bons) clients IMAP4 permettent de ne charger que les objets des messages, leur date et leur taille. Il est donc possible, même avec une mauvaise connexion, d'exploiter ce protocole sans blocage. Les conditions de travail ne sont plus les mêmes qu'avec le bon vieux POP3.

Comme nous sommes sur cette page surtout pour étudier le protocole, nous allons faire quelques

---

<sup>5</sup> FRC 2060 en français : <http://jlr31130.free.fr/rfc2060.html>

manips simples avec le serveur uw-imap installé sur gw2, avec un compte tout neuf : testimap2@gw2.maison.mrs. Il vient juste d'être créé et aucun client n'y a encore accédé.

## Préliminaires

Commençons par regarder ce qu'il y a dans le répertoire de l'utilisateur "testimap2" :

```
/home/testimap2# ls
/home/testimap2#
```

Il n'y a rien du tout.

## Ouverture d'une session IMAP

Il est possible d'utiliser la commande : LOGIN <user> <passwd>. Faisons-le :

```
c:\>telnet gw2.maison.mrs 143
* OK [CAPABILITY IMAP4REV1 X-NETSCAPE LOGIN-REFERRALS AUTH=LOGIN] gw2.maison.mrs
IMAP4rev1 2001.315 at Sat, 8 Nov 2003 17:16:42 +0100 (CET)
0001 LOGIN testimap2 testimap2
0001 OK [CAPABILITY IMAP4REV1 X-NETSCAPE NAMESPACE MAILBOX-REFERRALS SCAN SORT
THREAD=REFERENCES THREAD=ORDEREDSUBJECT MULTIAPPEND] User testimap2 authenticated
0002 LOGOUT
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
0002 OK LOGOUT completed
```

Ça marche. Dans un cas pareil, les identifiants du client passent en clair sur le réseau, comme avec POP3. Mais il est possible de faire mieux avec la commande AUTHENTICATE. Nous ne la verrons pas avec Telnet, vous comprendrez vite pourquoi. La manipulation est faite avec Thunderbird, le tout sniffé avec Ethereal :

Source	Destination	Protocol	Info
192.168.0.252	192.168.0.10	IMAP	Response: * OK [CAPABILITY IMAP4REV1 ...
192.168.0.10	192.168.0.252	IMAP	Request: 1 capability
192.168.0.252	192.168.0.10	IMAP	Response: * CAPABILITY IMAP4REV1 ...
192.168.0.10	192.168.0.252	IMAP	Request: 2 authenticate login
192.168.0.252	192.168.0.10	IMAP	Response: + VXNlciBOYW11AA==
192.168.0.10	192.168.0.252	IMAP	Request: dGVzdGltYXA=
192.168.0.252	192.168.0.10	IMAP	Response: + UGFzc3dvcmQA
192.168.0.10	192.168.0.252	IMAP	Request: dGVzdGltYXA=
192.168.0.252	192.168.0.10	IMAP	Response: 2 OK [CAPABILITY IMAP4REV1 ...

Vous voyez, c'est un peu compliqué à faire avec telnet. Avec cette méthode, c'est déjà un peu plus difficile de pirater les identifiants du client.

## Arranger son intérieur

Avec cette manipulation au moyen de Thunderbird, dont il n'est affiché qu'un court extrait, juste pour voir travailler "AUTHENTICATE", il s'est tout de même passé d'autres choses. La preuve, si nous retournons voir dans le répertoire de l'utilisateur :

```
/home/testimap2# ls -l
total 4
-rw----- 1 testimap2 nogroup 528 Nov 8 17:29 Trash
/home/testimap2#
```

Il y a maintenant un répertoire nommé "Trash". Il ne vient certainement pas du ciel, ce doit être Thunderbird qui l'a créé. Ressortons donc la trace Ethereal et cherchons...

```
Source      Destination  Protocol Info
...
192.168.0.10 192.168.0.252 IMAP      Request: 14 create "Trash"
192.168.0.252 192.168.0.10 IMAP      Response: 14 OK CREATE completed
...
```

Et nous trouvons ! La commande "CREATE" permet bien de créer un répertoire.

Essayons à la main :

```
c:\> telnet gw2.maison.mrs 143
* OK [CAPABILITY IMAP4REV1 ...
001 login testimap2 testimap2
001 OK [CAPABILITY IMAP4REV1 ...
002 create "Sent"
002 OK CREATE completed
003 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
003 OK LOGOUT completed
```

A priori, ça a l'air d'avoir marché. Vérification sur le serveur :

```
/home/testimap2# ls -l
total 8
-rw----- 1 testimap2 nogroup 528 Nov  8 18:01 Sent
-rw----- 1 testimap2 nogroup 528 Nov  8 17:29 Trash
/home/testimap2#
```

Pas de surprise.

Bien. Nous n'allons pas tout passer en revue, l'important est de comprendre le principe. Juste deux ou trois manips et ça ira bien.

## Lire un message

Pour ça, il faut déjà qu'il y en ait au moins un à lire. Envoyons un message par l'intermédiaire de "mail" directement depuis le serveur :

```
/home/testimap2# mail testimap2@gw2.maison.mrs
Subject: test simple
coucou
.
Cc:
/home/testimap2#
```

A-t-on ajouté quelque chose dans le répertoire de l'utilisateur ?

```
/home/testimap2# ls -l
total 8
-rw----- 1 testimap2 nogroup 528 Nov  8 18:01 Sent
-rw----- 1 testimap2 nogroup 528 Nov  8 17:29 Trash
/home/testimap2#
```

Il semble bien que non. Ça peut paraître curieux, mais c'est normal. Le message n'a pas encore été lu et il se trouve ailleurs. Vous aimeriez savoir où ? Nous l'avons déjà vu en page précédente, mais je vais vous le redire...

Nous utilisons sur cette machine le MTA Exim, qui range les message locaux dans le répertoire /var/spool/mail qui est en fait un alias de /var/mail :

```
/var/mail# ls -l
total 40
-rw-rw---- 1 chris mail 36189 Nov  8 12:41 chris
```

```
-rw-rw----  1 testimap2 mail      520 Nov  8 18:09 testimap2
/var/mail#
```

Ce fichier : "testimap2", contient tous les messages reçus. La preuve :

```
/var/mail# cat testimap2
From root@gw2.maison.mrs Sat Nov 08 18:09:47 2003
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by ca-marseille-34-108.w80-8.abo.wanadoo.fr with local (Exim 3.35 #1 (Debian))
        id lAIWap-0006ku-00
        for <testimap2@gw2.maison.mrs>; Sat, 08 Nov 2003 18:09:47 +0100
To: testimap2@gw2.maison.mrs
Subject: test simple
Message-Id: <E1AIWap-0006ku-00@ca-marseille-34-108.w80-8.abo.wanadoo.fr>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sat, 08 Nov 2003 18:09:47 +0100

coucou

/var/mail#
```

Allons faire des choses avec telnet :

```
* OK [CAPABILITY IMAP4REV1 ...
0001 login testimap2 testimap2
0001 OK [CAPABILITY IMAP4REV1 ...
0002 list "*" "*"
* LIST (\NoInferiors) NIL INBOX
* LIST (\NoInferiors \UnMarked) "/" Trash
* LIST (\NoInferiors \UnMarked) "/" .mailboxlist
* LIST (\NoInferiors \UnMarked) "/" Sent
* LIST (\NoInferiors) NIL INBOX
0002 OK LIST completed
0003 select INBOX
* 1 EXISTS
* NO Trying to get mailbox lock from process 26105
* 1 RECENT
* OK [UIDVALIDITY 1068314830] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/mail/testimap2
0003 OK [READ-WRITE] SELECT completed
0004 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
0004 OK LOGOUT completed
```

## La commande LIST

De la façon utilisée ici, elle ne fait qu'afficher le contenu du répertoire de l'utilisateur, à quelque chose près : INBOX, qui n'est rien d'autre que le fichier /var/mail/<user> dans notre exemple. Comme vous avez maintenant lu les RFC, vous savez que :

- NIL indique qu'il n'y a pas de "flag" particulier attribué à INBOX,
- \UnMarked signifie que le "dossier" ne contient pas de nouveaux messages depuis sa dernière sélection.
- .mailboxlist, rappelez-vous, n'est rien d'autre qu'un fichier texte qui contient la liste des dossiers créés dans le compte IMAP. Une commande "ls -la" le dévoile :

```
/home/testimap2# ls -la
total 20
drwxr-xr-x  2 testimap2 nogroup      4096 Nov  9 08:52 .
drwxrwsr-x  5 root      staff      4096 Nov  6 14:38 ..
-rw-r--r--  1 testimap2 nogroup      6 Nov  8 17:29 .mailboxlist
```

```
-rw----- 1 testimap2 nogroup 528 Nov 8 18:01 Sent
-rw----- 1 testimap2 nogroup 592 Nov 9 08:52 Trash
```

## La commande SELECT

Elle permet de sélectionner le dossier que l'on souhaite consulter. Dans l'exemple, nous apprenons :

- 1 EXISTS : il y a un message dedans,
- 1 RECENT : il y a un message nouveau depuis la dernière consultation.
- [UNSEEN 1] : il y a un message qui n'a pas été vu

Pour le reste, je vous laisse chercher dans les RFC.

Nous n'avons pas fait grand chose encore, mais retournons tout de même voir /var/mail/testimap2 :

```
/var/mail# cat testimap2
From root@gw2.maison.mrs Sat Nov 08 19:05:21 2003
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by ca-marseille-34-108.w80-8.abo.wanadoo.fr with local (Exim 3.35 #1 (Debian))
        id 1AIXSb-0006nG-00
        for <testimap2@gw2.maison.mrs>; Sat, 08 Nov 2003 19:05:21 +0100
To: testimap2@gw2.maison.mrs
Subject: test simple
Message-Id: <E1AIXSb-0006nG-00@ca-marseille-34-108.w80-8.abo.wanadoo.fr>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sat, 08 Nov 2003 19:05:21 +0100
X-IMAPbase: 1068314830 1
Status: 0
X-Status:
X-Keywords:
X-UID: 1
coucou
/var/mail#
```

A l'évidence, le serveur IMAP a rajouté quelques lignes dans l'en-tête du message...

Rejouons la connexion par telnet :

```
c:\> telnet gw2.maison.mrs 143
* OK [CAPABILITY IMAP4REV1 ...
01 login testimap2 testimap2
01 OK [CAPABILITY IMAP4REV1 ...
02 list "*" "*"
* LIST (\NoInferiors) NIL INBOX
* LIST (\NoInferiors \UnMarked) "/" Trash
* LIST (\NoInferiors \UnMarked) "/" .mailboxlist
* LIST (\NoInferiors \UnMarked) "/" Sent
* LIST (\NoInferiors) NIL INBOX
02 OK LIST completed
03 select INBOX
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1068365879] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
* OK [UNSEEN 1] first unseen message in /var/mail/testimap2
03 OK [READ-WRITE] SELECT completed
04 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
04 OK LOGOUT completed
```

La seule chose qui a changé, c'est que le "1 RECENT" est passé à "0 RECENT". Nous n'avons pas

lu le message (UNSEEN 1), mais le serveur a noté que depuis notre dernière visite, il n'y a pas eu de nouveaux messages.

Rien n'a changé dans /var/mail/testimap2.

## La commande FETCH

Vous avez pu constater dans les RFC la complexité de cette commande, nous allons l'utiliser ici simplement.

D'abord pour lire l'en-tête de l'unique message disponible (BODY[HEADER]) puis pour lire le texte du message (BODY[TEXT]) :

```
* OK [CAPABILITY IMAP4REV1...
001 login testimap2 testimap2
001 OK [CAPABILITY IMAP4REV1 ...
002 select INBOX
* 1 EXISTS
* NO Trying to get mailbox lock from process 28032
* 0 RECENT
* OK [UIDVALIDITY 1068367935] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
002 OK [READ-WRITE] SELECT completed

003 fetch 1 BODY[HEADER]
* 1 FETCH (BODY[HEADER] {471}
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by gw2.maison.mrs with local (Exim
3.35 #1 (Debian))
        id 1A1l1IA-0007Hi-00
        for <testimap2@gw2.maison.mrs>; Sun, 09 Nov 2003 09:51:30 +0100
To: testimap2@gw2.maison.mrs
Subject: test simple
Message-Id: <E1A1l1IA-0007Hi-00@gw2.maison.mrs>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sun, 09 Nov 2003 09:51:30 +0100
)
003 OK FETCH completed

004 fetch 1 BODY[TEXT]
* 1 FETCH (BODY[TEXT] {8}
coucou
)
004 OK FETCH completed
005 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
005 OK LOGOUT completed
```

Voyez les RFC pour une description complète des options de la commande FETCH.

Retournons voir dans /var/mail/testimap2 si quelque chose a changé :

```
/var/mail# cat testimap2
From root@gw2.maison.mrs Sun Nov 09 09:51:30 2003
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by gw2.maison.mrs with local (Exim 3.35 #1 (Debian))
        id 1A1l1IA-0007Hi-00
        for <testimap2@gw2.maison.mrs>; Sun, 09 Nov 2003 09:51:30 +0100
To: testimap2@gw2.maison.mrs
Subject: test simple
Message-Id: <E1A1l1IA-0007Hi-00@gw2.maison.mrs>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sun, 09 Nov 2003 09:51:30 +0100
```

```
X-IMAPbase: 1068314830 1
Status: RO
X-Status:
X-Keywords:
X-UID: 1

coucou

/var/mail#
```

Oui. Le "tag" Status est passé de O à RO.

## La commande STORE

Cette commande permet de modifier les "flags" attachés à un message.

Les flags que l'on peut attribuer à un message sont les suivants :

- \Seen            Le message a été lu
- \Answered      On a répondu au message
- \Flagged        Le message est "flagged" pour y donner une attention urgente/spéciale
- \Deleted        Le message est "deleted" (supprimé) pour que plus tard un EXPUNGE puisse l'enlever
- \Draft          Le message n'a pas été entièrement composé (marqué en tant que brouillon (draft)).
- \Recent         Le message est arrivé récemment dans cette boîte aux lettres. Cette session est la première session qui ait reçu une notification a propos de ce message.  
Les sessions ultérieures ne verront pas l'état \Recent pour ce message. Ce drapeau ne peut être modifié par le client.

S'il n'est pas possible de déterminer si oui ou non, cette session est la première session a être notifiée du message, alors ce message DEVRA (SHOULD) être considéré comme récent. Si de multiples connexions ont sélectionné la même boîte aux lettres simultanément, on ne peut définir laquelle de ces connexions verra les messages arrivés nouvellement avec l'état \Recent et quelles vont être celles qui le verront sans \Recent.

*Extrait de <http://jlr31130.free.fr/rfc2060.html#2.3.2>.*

Il est temps maintenant de reprendre un point **très important** de IMAP. Très important, parce que si l'on n'a pas compris ce qui va suivre, on va laisser son compte IMAP s'engraisser sans comprendre pourquoi et arrivera un jour où votre BAL se retrouvera pleine, vos messages entrants seront refusés, alors que vous pensez avoir bien fait le ménage.

**Un message considéré comme effacé ne l'est pas.** Il a juste le flag \Deleted positionné.

Comme c'est clairement indiqué, seul le flag "\Recent" ne peut être modifié par le client.

Le problème qui se pose avec la plupart des clients de messagerie est le suivant :

- Ces clients créent une poubelle (répertoire "Trash", avec Thunderbird),
- l'effacement d'un message dans INBOX se traduit la plupart du temps par deux opérations :

- le message est **marqué** \Deleted dans INBOX,
- le message est **copié** dans Trash.

Lorsque l'utilisateur méthodique efface ensuite le contenu de la poubelle, il ne fait que marquer dans Trash les messages avec le flag \Deleted.

Vu de dehors, tout semble vide, vu de dedans, votre message existe toujours, et en double, en plus !

## Démonstration

Nous avons déjà vu ça en page précédente, mais c'est tellement important qu'il vaut mieux le répéter

Nous repartons d'un compte IMAP parfaitement vide, nous le vérifions sur le serveur :

```
/var/mail# cat testimap2
# Il n'y a rien dans /var/mail/testimap2

/var/mail# cat /home/testimap2/Trash
From MAILER-DAEMON Sun Nov 9 10:48:15 2003
Date: 09 Nov 2003 10:48:15 +0100
From: Mail System Internal Data <MAILER-DAEMON@gw2.maison.mrs>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
Message-ID: <1068371295@gw2.maison.mrs>
X-IMAP: 1068308988 0000000004
Status: RO

This text is part of the internal format of your mail folder, and is not
a real message. It is created automatically by the mail system software.
If deleted, important folder data will be lost, and it will be re-created
with the data reset to initial values.

# Là, il y a quelque chose, mais le texte du message l'indique clairement:
# c'est un message nécessaire au système MAILBOX , ce n'est pas un réel message
# et il ne faut pas le détruire.

/var/mail#
```

Envoi d'un message de test, comme vu plus haut :

```
/var/mail# mail testimap2@gw2.maison.mrs
Subject: test DELETE
message destiné à devenir un fantôme...
.
Cc:
ca-marseille-35-89:/var/mail#
```

Ce n'est pas la peine de tout refaire, nous savons qu'il est maintenant dans /var/mail/testimap2.

Nous allons utiliser Thunderbird pour :

- Le lire,
- l'effacer dans Inbox, donc le copier dans la poubelle (Trash),
- l'effacer de la poubelle.

Mais Ethereal va suivre toutes ces opérations...

Pour alléger la lecture, je supprime la partie authentification, repérage de tous les paramètres IMAP, de même que tout ce qui ne concerne que TCP. Nous démarrons à la sélection de INBOX :

```
Source      Destination  Protocol Info
# Sélection de INBOX
192.168.0.15 192.168.0.252 IMAP Request: 13 select "INBOX"
192.168.0.252 192.168.0.15 IMAP Response: * 1 EXISTS
```



```

192.168.0.15 192.168.0.252 IMAP Request: 14 UID fetch 1:* (FLAGS)
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (UID 1 FLAGS (\Recent))
192.168.0.15 192.168.0.252 IMAP Request: 15 UID fetch 1 (UID RFC822.SIZE FLAGS
BODY.PEEK[HEADER...]
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (UID 1 RFC822.SIZE 512 FLAGS (\Recent)
BODY[HEADER.FIELDS...]
# Lecture du message
192.168.0.15 192.168.0.252 IMAP Request: 16 UID fetch 1 (UID RFC822.SIZE BODY[])
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (UID 1 RFC822.SIZE 512 BODY[] {512})
# Copie dans Trash
192.168.0.15 192.168.0.252 IMAP Request: 17 uid copy 1 "Trash"
192.168.0.252 192.168.0.15 IMAP Response: 17 OK UID COPY completed
# Effacement de INBOX (positionnement du flag "\Deleted")
192.168.0.15 192.168.0.252 IMAP Request: 18 uid store 1 +FLAGS (\Deleted)
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (FLAGS (\Recent \Seen \Deleted) UID 1)
192.168.0.252 192.168.0.15 IMAP Response: * OK [CAPABILITY IMAP4REV1 ...
...
# Sélection de Trash
192.168.0.15 192.168.0.252 IMAP Request: 2 select "Trash"
192.168.0.252 192.168.0.15 IMAP Response: * 1 EXISTS
192.168.0.15 192.168.0.252 IMAP Request: 3 UID fetch 1:* (FLAGS)
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (UID 6 FLAGS (\Seen))
192.168.0.15 192.168.0.252 IMAP Request: 4 UID fetch 6 (UID RFC822.SIZE FLAGS
BODY.PEEK[HEADER...]
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (UID 6 RFC822.SIZE 512 FLAGS (\Seen)
BODY[HEADER.FIELDS...]
# Lecture du message qui se trouve dans la poubelle
192.168.0.15 192.168.0.252 IMAP Request: 5 UID fetch 6 (UID RFC822.SIZE BODY[])
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (UID 6 RFC822.SIZE 512 BODY[] {512})
# Effacement du message dans Trash (positionnement du flag \Deleted)
192.168.0.15 192.168.0.252 IMAP Request: 6 uid store 6 +Flags (\Deleted)
192.168.0.252 192.168.0.15 IMAP Response: * 1 FETCH (FLAGS (\Seen \Deleted) UID 6)
...
# Il ne se passe plus rien d'important.

```

A ce niveau de la manipulation, l'utilisateur de Thunderbird :

- ne trouve plus rien dans Inbox,
- ne trouve plus rien dans Trash,
- croit donc que son système de messagerie est complètement vide et propre...

Il n'en est rien, la preuve :

```

/var/mail# cat testimap2
From root@gw2.maison.mrs Sun Nov 09 11:03:25 2003
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by gw2.maison.mrs with local (Exim 3.35 #1 (Debian))
id 1AImP1-0007KO-00
for <testimap2@gw2.maison.mrs>; Sun, 09 Nov 2003 11:03:25 +0100
To: testimap2@gw2.maison.mrs
Subject: test DELETE
Message-Id: <E1AImP1-0007KO-00@gw2.maison.mrs>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sun, 09 Nov 2003 11:03:25 +0100
X-IMAPbase: 1068372272 1
Status: RO
X-Status: D
X-Keywords:
X-UID: 1

message destiné à devenir un fantôme...

/var/mail#

```

Dans /var/mail/testimap2 (INBOX) le message est toujours présent, il n'y a que **X-STATUS: D** pour indiquer qu'il est détruit.

```

/var/mail# cat /home/testimap2/Trash
From MAILER-DAEMON Sun Nov  9 11:04:51 2003
Date: 09 Nov 2003 11:04:51 +0100
From: Mail System Internal Data <MAILER-DAEMON@gw2.maison.mrs>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
Message-ID: <1068372291@gw2.maison.mrs>
X-IMAP: 1068308988 0000000006
Status: RO

This text is part of the internal format of your mail folder, and is not
a real message.  It is created automatically by the mail system software.
If deleted, important folder data will be lost, and it will be re-created
with the data reset to initial values.

From root@gw2.maison.mrs Sun Nov 09 11:03:25 2003
Return-path: <root@gw2.maison.mrs>
Envelope-to: testimap2@gw2.maison.mrs
Received: from root by gw2.maison.mrs with local (Exim 3.35 #1 (Debian))
        id 1AImPl-0007KO-00
        for <testimap2@gw2.maison.mrs>; Sun, 09 Nov 2003 11:03:25 +0100
To: testimap2@gw2.maison.mrs
Subject: test DELETE
Message-Id: <E1AImPl-0007KO-00@gw2.maison.mrs>
From: Christian Caleca <root@gw2.maison.mrs>
Date: Sun, 09 Nov 2003 11:03:25 +0100
Status: RO
X-Status: D
X-Keywords:
X-UID: 6

message destiné à devenir un fantôme...

/var/mail#

```

Dans /home/testimap2/Trash, la poubelle, **le message est toujours présent**, il n'y a que **X-STATUS: D** pour indiquer qu'il est détruit.

Autrement dit, alors même que l'on croit s'être définitivement débarrassé du message, on n'a fait que le copier en double dans son système de messagerie...

Comment faire alors ?

## La commande EXPUNGE

Rassurez vous, IMAP4rev1 a prévu cet ennui et met à disposition une commande qui élimine définitivement tous les messages marqués \Deleted dans un répertoire donné.

Nous l'utilisons avec Telnet :

```

* OK [CAPABILITY IMAP4REV1...
001 login testimap2 testimap2
001 OK [CAPABILITY IMAP4REV1 ...
002 select INBOX
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1068372272] UID validity status
* OK [UIDNEXT 2] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
002 OK [READ-WRITE] SELECT completed
003 expunge
* 1 EXPUNGE
* 0 EXISTS
* 0 RECENT
003 OK Expunged 1 messages

004 select Trash

```

```
* 1 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1068308988] UID validity status
* OK [UIDNEXT 7] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
004 OK [READ-WRITE] SELECT completed
005 expunge
* 1 EXPUNGE
* 0 EXISTS
* 0 RECENT
005 OK Expunged 1 messages
006 logout
* BYE gw2.maison.mrs IMAP4rev1 server terminating connection
006 OK LOGOUT completed
```

retour sur le serveur :

```
/var/mail# cat testimap2
/var/mail#
```

/var/mail/testimap2 (INBOX) est bien vide...

```
/var/mail# cat /home/testimap2/Trash
From MAILER-DAEMON Sun Nov 9 11:40:30 2003
Date: 09 Nov 2003 11:40:30 +0100
From: Mail System Internal Data <MAILER-DAEMON@gw2.maison.mrs>
Subject: DON'T DELETE THIS MESSAGE -- FOLDER INTERNAL DATA
Message-ID: <1068374430@gw2.maison.mrs>
X-IMAP: 1068308988 0000000006
Status: RO

This text is part of the internal format of your mail folder, and is not
a real message. It is created automatically by the mail system software.
If deleted, important folder data will be lost, and it will be re-created
with the data reset to initial values.

/var/mail#
```

et /home/testimap2/Trash est également vide. Ouf !

## Conclusions

Ce court exposé n'avait d'autre ambition que de montrer quelques points importants. Il est clair que celui qui voudra développer un client de messagerie IMAP devra effectivement lire les RFC et faire beaucoup plus de manipulations préliminaires que celles que nous avons vues ici.

## Emploi des commandes IMAP

Les commande IMAP sont toutes en mode texte, comme pour tout protocole d'application "classique". Elles sont donc utilisables plus ou moins simplement avec telnet. Ici, c'est nettement plus compliqué qu'avec POP3, mais ça reste faisable.

### *Pourquoi sont-elles précédées d'un "tag" ?*

Comme vous avez attentivement lu les RFC, vous savez que c'est parce que le client peut envoyer plusieurs commandes sans obligatoirement attendre à chaque fois la réponse. Le "tag" permet donc de retrouver facilement la réponse à une commande donnée.

### *De l'importance de EXPUNGE...*

Nous avons vu qu'il est fondamental de paramétrer correctement son client de messagerie pour qu'il envoie périodiquement la commande EXPUNGE au serveur sur les divers dossiers de notre messagerie. Avec Thunderbird :

- il est possible de la faire automatiquement en certaines occasions, fouillez dans les diverses options de configuration du client,
- il est possible de le faire manuellement, en sélectionnant un dossier, puis en cliquant du bouton droit dessus et en sélectionnant "Compact This Folder".

Je vous laisse le soin de trouver l'équivalent sur d'autres clients de messagerie.

## **Le format MAILBOX**

Au travers de ces manipulations, nous avons pu comprendre que le format MAILBOX consiste en un unique fichier par dossier, dans lequel tous les messages sont ajoutés les uns derrière les autres, avec quelques drapeaux spécifiques pour indiquer l'état de ces messages (X-IMAPbase:, Status:, X-Status:, X-Keywords:, X-UID: ).

Ce système, d'ailleurs repris par la plupart des clients de messagerie pour le stockage en local des messages, offre au moins un gros inconvénient : si le fichier est endommagé, la totalité de son contenu sera probablement perdue.

Le format MAILDIR, que nous n'avons pas vu ici, élimine en grande partie cet inconvénient. Nous le verrons rapidement dans la page suivante.

## Deux serveurs IMAP4

Si tout ça vous a donné l'envie d'expérimenter, voire de mettre en place un serveur IMAP, voici en quelque mots une présentation des deux serveurs les plus courants.

### Le cas simple et facile à installer

Il s'agit du serveur uw-imapd, celui là même qui a été utilisé en page précédente. Utilisé sur Debian ou une autre distribution comme Mandrake ou Fedora, avec Exim, Postfix ou Sendmail, les MTA installés par défaut savent délivrer localement avec les bons outils dans des boîtes aux lettres au format Mailbox, généralement dans le répertoire /var/spool/mail.

#### Pour Mandrake

Installez le paquetage imap sur Mandrake (celui-ci vous fournira également le service POP3).

Vérifiez que le super démon xinetd est correctement configuré. Vous devez trouver dans /etc/xinetd.d/imap quelque chose de ce genre :

```
service imap
{
    socket_type      = stream
    wait            = no
    user            = root
    server          = /usr/sbin/imapd
    log_on_success  += DURATION USERID
    log_on_failure  += USERID
    disable         = no
}
```

Vérifiez également dans /etc/services la présence de ces lignes :

```
imap      143/tcp      imap2      # Interim Mail Access Proto v2
imap      143/udp      imap2
```

Si imap2 vous gêne, remplacez par imap4. Ce sera plus joli, mais ça ne fonctionnera pas mieux.

Curieusement, imap2 fait en réalité référence à imap4 révision 1

#### Pour Debian :

Installez le paquetage uw-imapd.

Vérifiez que le super démon inetd est correctement configuré. Vous devez trouver dans /etc/inetd.conf quelque chose de ce genre :

```
#:MAIL: Mail, news and uucp services.
imap2 stream tcp      nowait      root       /usr/sbin/tcpd  /usr/sbin/imapd
imap3 stream tcp      nowait      root       /usr/sbin/tcpd  /usr/sbin/imapd
```

Vérifiez également dans /etc/services la présence de ces lignes :

```
imap2      143/tcp      imap      # Interim Mail Access Proto v2
imap2      143/udp      imap
```

## Pour les deux :

relancez votre super démon si vous avez modifié sa configuration, et ça devrait fonctionner.

Vous voyez, ce n'est pas bien compliqué, et ce sera largement suffisant dans bien des cas.

## Le cas compliqué et difficile à installer

Les solutions les plus simples n'étant pas forcément les plus attrayantes, nous allons maintenant voir une solution qui utilise Cyrus, le serveur de choc.

- MTA : Postfix
- IMAP : Cyrus en version 2.1
- Authentification des utilisateurs par saslauth.

## Pourquoi tout ça ?

L'objectif est de monter un système indépendant des comptes d'utilisateurs UNIX (Authentification par saslauth via une base de données indépendante des comptes utilisateurs), avec un serveur IMAP proposant le plus de fonctionnalités possibles, et utilisant le format Maildir, plus sûr (Cyrus).

Pour l'authentification, nous aurions pu utiliser une base de données de type MySQL ou un annuaire LDAP. Le sado-masochisme a toutefois ses limites, et ça nous mènerait trop loin hors du sujet initial.

*Un outil comme web-cyradm<sup>6</sup> propose une solution en utilisant MySQL. Cet outil, pour prometteur qu'il soit, ne semble pas encore assez mature. Il n'est pas le seul dans ce genre, replex<sup>7</sup> semble être un concurrent très proche.*

*Avec ce trio, nous aurions pu réaliser un système de messagerie performant :*

- *Administrable par une interface web (web-cyradm),*
- *capable de créer des boîtes aux lettres pour des domaines virtuels (des domaines autres que celui auquel appartient le serveur),*
- *capable de gérer les quotas pour chaque boîte, sans passer par les quotas des utilisateurs UNIX,*
- *la possibilité de gérer les redirections et les répondeurs,*
- *gérer de multiples alias pour une même boîte aux lettres,*
- *gérer un compte "catch all" c'est à dire un collecteur de messages destinés à votre domaine, mais à des utilisateurs qui n'existent pas.*

*Mais attention, ça peut vite devenir très compliqué...*

## Matériel requis

Ici, nous nous contenterons de faire fonctionner cyrus d'une façon "simple".

---

<sup>6</sup> web-cyradm : <http://www.web-cyradm.org/>

<sup>7</sup> replex : <http://sourceforge.net/projects/replex/>

Les paquetages existent pour la Mandrake 9.2 dans les contributions. Vous pouvez faire ça sur Mandrake, mais ce qui suit est décrit sur Debian. Ce sera probablement plus compliqué, mais par la suite, c'est tout de même plus facile de faire évoluer une Debian qu'une Mandrake.

Il faut la version 2.1 ou supérieure de Cyrus. Elle n'existe pas "packagée" dans la Debian stable. Autrement dit vous avez le choix entre :

- Compiler sur votre version stable le paquetage source "testing"
- utiliser la Debian "testing"

Nous ferons ça sur une testing.

Comme il n'est pas question ici d'écrire une encyclopédie, nous supposons que vous savez faire les choses suivantes :

- installer une Debian,
- la passer en version "testing",
- installer Postfix à la place d'Exim (Si vous êtes un expert d'Exim, gardez Exim. Il faut juste être capable de faire comprendre à Exim qu'il doit utiliser Cyrus pour le transport local,
- savoir en gros comment fonctionne PAM (Pluggable Authentication Modules).

Si vous savez faire tout cela, vous pourrez faire aussi la suite. Sinon, ça risque de se solder par un échec.

## Cyrus

C'est lui qui va recevoir les mails locaux, gérer les boîtes aux lettres des inscrits et leur servir leurs messages via IMAP (ou POP3).

Cyrus, pour authentifier les clients, s'appuie sur SASL. SASL peut authentifier depuis par plusieurs méthodes :

- sasldb, une base de données au format Berkeley,
- shadow, en utilisant les comptes UNIX locaux,
- pam, en utilisant à peu près n'importe quoi.

Dans le cas le plus simple, shadow, chaque utilisateur devra disposer d'un compte local, ce n'est pas ce qui nous intéresse.

sasldb, c'est déjà mieux, les utilisateurs auront un compte dans la base sasldb, indépendant des comptes UNIX,

pam, c'est le moyen le plus souple. On pourra utiliser un annuaire LDAP ou une base de données MySQL ou même sasldb, via pam.

La première chose à faire, une fois la configuration vue plus haut réalisée, est d'installer Cyrus21 et saslauthd.

## Installation de Cyrus :

Voici la liste des paquetages. Attention, à ceux qui sont installés (ii). Tous ceux qui sont listés ici ne

sont pas forcément nécessaires.

```
cyclope:~# dpkg -l '*cyrus*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
|/ Name Version Description
+++-----
```

Name	Version	Description
pn cyrus-admin	<none>	(no description available)
pn cyrus-common	<none>	(no description available)
pn cyrus-dev	<none>	(no description available)
pn cyrus-imapd	<none>	(no description available)
pn cyrus-nntp	<none>	(no description available)
pn cyrus-pop3d	<none>	(no description available)
ii cyrus21-admin	2.1.14-1	Cyrus mail system (administration tool)
ii cyrus21-clients	2.1.14-1	Cyrus mail system (test clients)
ii cyrus21-common	2.1.14-1	Cyrus mail system (common files)
pn cyrus21-dev	<none>	(no description available)
ii cyrus21-doc	2.1.14-1	Cyrus mail system (documentation files)
un cyrus21-docs	<none>	(no description available)
ii cyrus21-imapd	2.1.14-1	Cyrus mail system (IMAP support)
pn cyrus21-murder	<none>	(no description available)
ii cyrus21-pop3d	2.1.14-1	Cyrus mail system (POP3 support)
un cyrus22-common	<none>	(no description available)
ii libauthen-sasl-cyrus-per	0.07-1	Perl extension for Cyrus SASL library
un libcyrus-imap-perl	<none>	(no description available)
ii libcyrus-imap-perl21	2.1.14-1	Interface to Cyrus imap client imclient library

## installation de sasl :

```
cyclope:~# dpkg -l '*sasl*'
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Installed/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
|/ Name Version Description
+++-----
```

Name	Version	Description
pn gsasl	<none>	(no description available)
ii libauthen-sasl-cyrus-per	0.07-1	Perl extension for Cyrus SASL library
ii libauthen-sasl-perl	2.05-1	Authen::SASL - SASL Authentication framework
pn libgsasl1	<none>	(no description available)
pn libgsasl1-dev	<none>	(no description available)
un libsasl-bin	<none>	(no description available)
pn libsasl-dev	<none>	(no description available)
pn libsasl-digestmd5-des	<none>	(no description available)
pn libsasl-digestmd5-plain	<none>	(no description available)
pn libsasl-gssapi-heimdal	<none>	(no description available)
pn libsasl-gssapi-mit	<none>	(no description available)
pn libsasl-krb4-mit	<none>	(no description available)
un libsasl-modules	<none>	(no description available)
un libsasl-modules-krb5	<none>	(no description available)
un libsasl-modules-nonus	<none>	(no description available)
ii libsasl-modules-plain	1.5.27-3.5	Basic Pluggable Authentication Modules for SASL
ii libsasl2	2.1.12-1	Authentication abstraction library
pn libsasl2-dev	<none>	(no description available)
un libsasl2-digestmd5-des	<none>	(no description available)
pn libsasl2-digestmd5-plain	<none>	(no description available)
pn libsasl2-gssapi-mit	<none>	(no description available)
pn libsasl2-krb4-mit	<none>	(no description available)
ii libsasl2-modules	2.1.12-1	Pluggable Authentication Modules for SASL
ii libsasl2-modules-gssapi	2.1.12-1	Pluggable Authentication Modules for SASL
ii libsasl2-modules-kerbero	2.1.12-1	Pluggable Authentication Modules for SASL
pn libsasl2-modules-plain	<none>	(no description available)
ii libsasl7	1.5.27-3.5	Authentication abstraction library.
ii sasl-bin	1.5.27-3.5	Programs for manipulating the SASL users database
ii sasl2-bin	2.1.12-1	Programs for manipulating the SASL users database



Là encore, tous ne sont pas forcément nécessaires.

## Vérifications

### saslauthd

Le paquetage sasl2-bin contient les outils nécessaires pour gérer l'authentification sasl :

```
cyclope:~# dpkg -L sasl2-bin
/.
/usr
/usr/share
/usr/share/man
/usr/share/man/man8
/usr/share/man/man8/sasldblistusers2.8.gz
/usr/share/man/man8/saslpasswd2.8.gz
/usr/share/man/man8/sasldbconverter2.8.gz
/usr/share/man/man8/sasltestsuite.8.gz
/usr/share/man/man8/saslauthd.8.gz
/usr/share/man/man8/testsaslauthd.8.gz
/usr/share/doc
/usr/share/doc/sasl2-bin
/usr/share/doc/sasl2-bin/README.Debian
/usr/share/doc/sasl2-bin/copyright
/usr/share/doc/sasl2-bin/changelog.gz
/usr/share/doc/sasl2-bin/changelog.Debian.gz
/usr/sbin
/usr/sbin/saslauthd
/usr/sbin/sasltestsuite
/usr/sbin/testsaslauthd
/usr/sbin/sasldblistusers2
/usr/sbin/saslpasswd2
/usr/sbin/sasldbconverter2
/etc
/etc/default
/etc/default/saslauthd
/etc/init.d
/etc/init.d/saslauthd
```

- saslauthd, c'est le "daemon",
- saslpasswd2 permet de créer des couples utilisateur/mot de passe,
- sasldblistusers2 permet de lister les utilisateurs,
- testsaslauthd permet de tester un "login".

Le fichier /etc/default/saslauthd a son importance :

```
cyclope:~# cat /etc/default/saslauthd
# This needs to be uncommented before saslauthd will be run automatically
START=yes

# You must specify the authentication mechanisms you wish to use.
# This defaults to "pam" for PAM support, but may also include
# "shadow" or "sasldb", like this:
# MECHANISMS="pam shadow"
MECHANISMS="sasldb"
```

- START=yes est primordial si vous voulez que le "daemon" soit chargé via SysV,
- MECHANISMS définit le mécanisme utilisé par saslauthd pour authentifier. Ici, nous utiliserons sasldb.

C'est parti...

- Démarrage de saslauthd :

```
# /etc/init.d/saslauthd restart
Restarting SASL Authentication Daemon: saslauthd.
#
```

- Création d'un compte :

```
# saslpasswd2 -c cyrus
Password:
Again (for verification):
#
```

Nous avons créé un utilisateur nommé cyrus, avec un mot de passe : passwd dans l'exemple. Bien entendu, essayez de choisir quelque chose de plus original. Cet utilisateur est destiné à devenir par la suite l'administrateur de Cyrus.

- Vérifications :

```
# sasldblistusers2
cyrus@cyclope: userPassword
#
```

l'utilisateur cyrus est bien créé (cyclope est le nom de la machine de test).

```
# testsaslauthd -u cyrus -p passwd
0: OK "Success."
cyclope:/etc/pam.d#
```

saslauthd semble fonctionner correctement. Nous pouvons passer à cyrus.

## Cyrus

Le paquetage cyrus21-admin contient :

```
# dpkg -L cyrus21-admin
/.
/usr
/usr/share
/usr/share/linda
/usr/share/linda/overrides
/usr/share/linda/overrides/cyrus21-admin
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/cyrus21-admin
/usr/share/icons
/usr/share/icons/cyradm.xpm
/usr/share/icons/mini
/usr/share/icons/mini/cyradm.xpm
/usr/share/doc
/usr/share/doc/cyrus21-admin
/usr/share/doc/cyrus21-admin/README.Debian
/usr/share/doc/cyrus21-admin/copyright
/usr/share/doc/cyrus21-admin/changelog.gz
/usr/share/doc/cyrus21-admin/changelog.Debian.gz
/usr/share/doc/cyrus21-admin/changelog.html.gz
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/cyradm.1.gz
/usr/share/man/man1/sieveshell.1.gz
/usr/bin
/usr/bin/cyradm
```

```
/usr/bin/sieveshell
/usr/lib
/usr/lib/menu
/usr/lib/menu/cyrus21-admin
```

cyradm est l'outil qui permettra de gérer les comptes imap.

Le paquetage cyrus21-imapd contient :

```
cyclope:/etc/pam.d# dpkg -L cyrus21-imapd
/.
/usr
/usr/sbin
/usr/sbin/squatter
/usr/lib
/usr/lib/cyrus
/usr/lib/cyrus/bin
/usr/lib/cyrus/bin/imapd
/usr/share
/usr/share/man
/usr/share/man/man8
/usr/share/man/man8/squatter.8.gz
/usr/share/man/man8/idled.8.gz
/usr/share/man/man8/imapd.8.gz
/usr/share/doc
/etc
/etc/pam.d
/etc/pam.d/imap
/usr/share/doc/cyrus21-imapd
```

- le "daemon" imapd, qui sera en fait piloté à travers le "super daemon" (inetd ou xinetd),
- un fichier de configuration pour pam. Il ne nous intéresse pas pour l'instant.

Quant au paquetage cyrus21-common :

```
cyclope:/etc/pam.d# dpkg -L cyrus21-common
/.
/etc
/etc/pam.d
/etc/pam.d/lmtp
/etc/pam.d/sieve
/etc/default
/etc/default/cyrus21
/etc/logcheck
/etc/logcheck/ignore.d.server
/etc/logcheck/ignore.d.server/cyrus21
/etc/logcheck/ignore.d.workstation
/etc/logcheck/ignore.d.workstation/cyrus21
/etc/cyrus.conf
/etc/imapd.conf
/etc/init.d
/etc/init.d/cyrus21
/etc/cron.daily
/etc/cron.daily/cyrus21-common
/usr
/usr/sbin
/usr/sbin/cyrdeliver
/usr/sbin/cyrdump
/usr/sbin/cyrmaster
/usr/sbin/cyrquota
/usr/sbin/cyrreconstruct
/usr/sbin/cyrus-makedirs
/usr/sbin/ctl_cyrusdb
/usr/sbin/ctl_deliver
/usr/sbin/ctl_mboxlist
/usr/sbin/cvt_cyrusdb
/usr/sbin/chk_cyrus
/usr/sbin/ipurge
/usr/sbin/tls_prune
```

```
/usr/sbin/mbpath
/usr/lib
/usr/lib/cyrus
/usr/lib/cyrus/upgrade
/usr/lib/cyrus/upgrade/convert-sieve.pl
/usr/lib/cyrus/upgrade/dohash
/usr/lib/cyrus/upgrade/rehash
/usr/lib/cyrus/upgrade/undohash
/usr/lib/cyrus/upgrade/translatesieve
/usr/lib/cyrus/upgrade/upgradesieve
/usr/lib/cyrus/cyrus-db-types.txt
/usr/lib/cyrus/cyrus-hardwired-config.txt
/usr/lib/cyrus/bin
/usr/lib/cyrus/bin/lmtpd
/usr/lib/cyrus/bin/timsieved
/usr/lib/cyrus/bin/notifyd
/usr/lib/cyrus/bin/fud
/usr/lib/cyrus/get-backtrace.gdb
/usr/share
/usr/share/snmp
/usr/share/snmp/mibs
/usr/share/snmp/mibs/CYRUS-MASTER-MIB.txt
/usr/share/linda
/usr/share/linda/overrides
/usr/share/linda/overrides/cyrus21-common
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/cyrus21-common
/usr/share/doc
/usr/share/doc/cyrus21-common
/usr/share/doc/cyrus21-common/README.Debian.gz
/usr/share/doc/cyrus21-common/README.Debian.debug.gz
/usr/share/doc/cyrus21-common/README.Debian.simpleinstall.gz
/usr/share/doc/cyrus21-common/README.postfix.gz
/usr/share/doc/cyrus21-common/changelog.Debian.gz
/usr/share/doc/cyrus21-common/copyright
/usr/share/doc/cyrus21-common/NEWS.Debian.gz
/usr/share/doc/cyrus21-common/changelog.gz
/usr/share/doc/cyrus21-common/UPGRADE.Debian.gz
/usr/share/doc/cyrus21-common/README.Debian.database.gz
/usr/share/doc/cyrus21-common/changelog.html.gz
/usr/share/man
/usr/share/man/man5
/usr/share/man/man5/cyrus.conf.5.gz
/usr/share/man/man5/imapd.conf.5.gz
/usr/share/man/man8
/usr/share/man/man8/mbpath.8.gz
/usr/share/man/man8/timsieved.8.gz
/usr/share/man/man8/notifyd.8.gz
/usr/share/man/man8/lmtpd.8.gz
/usr/share/man/man8/ctl_cyrusdb.8.gz
/usr/share/man/man8/ctl_deliver.8.gz
/usr/share/man/man8/ctl_mboxlist.8.gz
/usr/share/man/man8/cvt_cyrusdb.8.gz
/usr/share/man/man8/chk_cyrus.8.gz
/usr/share/man/man8/fud.8.gz
/usr/share/man/man8/ipurge.8.gz
/usr/share/man/man8/cyrdeliver.8.gz
/usr/share/man/man8/cyrmaster.8.gz
/usr/share/man/man8/cyrreconstruct.8.gz
/usr/share/man/man8/cyrquota.8.gz
/usr/share/man/man8/cyrdump.8.gz
/usr/share/man/man8/cyrus-makedirs.8.gz
/usr/share/man/man8/tls_prune.8.gz
/var
/var/lib
/var/lib/cyrus
/var/spool
/var/spool/cyrus
/var/spool/sieve
/var/run
/var/run/cyrus
```

```
/var/run/cyrus/socket
/etc/sieve
/usr/lib/cyrus/bin/quota
/usr/lib/cyrus/bin/reconstruct
```

- les fichiers de configuration de cyrus (/etc/cyrus.conf) et de imapd (/etc/imapd.conf),
- le script pour initialiser cyrus (/etc/init.d/cyrus21)

Il n'y a à priori rien à modifier dans la configuration par défaut de cyrus. Pour imapd, ça va être différent.

Le man imapd.conf vous donnera tous les détails, pour l'instant, nous devons surtout prêter attention à trois lignes :

```
admins: cyrus
```

Il faut indiquer qui sera le (ou les) administrateur(s) de cyrus imapd. Autant choisir tout de suite le compte que nous avons créé tout à l'heure, parce que la seconde ligne qui nous intéresse :

```
sasl_pwcheck_method: saslauthd
```

indique la méthode d'authentification. Ce sera obligatoirement SASL, mais SASL peut utiliser divers moyens, nous choisissons saslauthd, puisque nous l'avons testée avec succès.

Enfin :

```
servername: cyclope.maison.mrs
```

servername doit indiquer le nom complètement qualifié de l'hôte de cyrus.

Bien. Cyradm devrait fonctionner.

- Réinitialisons cyrus :

```
# /etc/init.d/cyrus21 restart
Stopping Cyrus IMAPd: cyrmaster.
Waiting for complete shutdown...
Starting Cyrus IMAPd: cyrmaster.
```

- démarrons cyradm avec l'utilisateur cyrus (déclaré comme administrateur) :

```
cyclope:/etc/pam.d# cyradm --user cyrus localhost
IMAP Password:

cyclope>
```

Ben voilà... Heureusement, il y a une aide en ligne :

```
cyclope> help
authenticate, login, auth      authenticate to server
chdir, cd                      change current directory
createmailbox, create, cm     create mailbox
deleteaclmailbox, deleteacl, dam remove ACLs from mailbox
deletemailbox, delete, dm     delete mailbox
disconnect, disc              disconnect from current server
exit, quit                    exit cyradm
help, ?                        show commands
info                           display mailbox/server metadata
listacl, lam, listaclmailbox  list ACLs on mailbox
listmailbox, lm               list mailboxes
listquota, lq                 list quotas on specified root
listquotaroot, lqr, lqm      show quota roots and quotas for mailbox
```

```

reconstruct                reconstruct mailbox (if supported)
renamemailbox, rename, renm rename (and optionally relocate) mailbox
server, servername, connect show current server or connect to server
setaclmailbox, sam, setacl  set ACLs on mailbox
setinfo                    set server metadata
setquota, sq               set quota on mailbox or resource
version, ver                display version info of current server
cyclope>

```

commençons par un "listmailbox" :

```

cyrus> listmailbox
cyrus>

```

C'est vide. C'est normal, nous n'avons encore rien fait.

```

cyclope> createmailbox user.essai0
cyclope> lm
user.essai0 (\HasNoChildren)
cyclope>

```

en voilà un de fait. Voyons les droits :

```

cyclope> listaclmailbox user.essai0
essai0 lrswipcda
cyclope>

```

seul `essai0` a tous les droits sur cette boîte aux lettres. D'ailleurs :

```

cyclope> deletemailbox user.essai0
deletemailbox: Permission denied
cyclope>

```

même l'administrateur cyrus ne peut détruire cette boîte. Du moins, pas si facilement.

Faisons un petit tour sur le serveur, pour voir. L'installation de cyrus a construit toute une arborescence dans `/var/spool/cyrus`. Je vous passe les détails, pour aller directement dans :

```

cyclope:/var/spool/cyrus/mail/e/user/essai0# ls
cyrus.cache cyrus.header cyrus.index
cyclope:/var/spool/cyrus/mail/e/user/essai0#

```

Comme c'est peut-être pas clair, détaillons un peu.

- Jusqu'à `/var/spool/cyrus/mail/e/` c'est l'installation de cyrus qui a créé l'arborescence. Vous trouverez un répertoire par lettre de l'alphabet, ils sont tous vides, sauf le "e".
- Dans le "e", `user/essai0` a bien entendu été créé lorsque l'on a créé la BAL avec cyradm. Les trois fichiers inclus également.

## Qu'avons-nous fait ?

- Nous avons mis en place un service d'authentification, basé sur SASL,
- nous avons créé un compte d'administration pour IMAP : cyrus,

- nous avons créé une BAL de test: [essai0@cyrus.maison.mrs](mailto:essai0@cyrus.maison.mrs).

## Que reste-t-il à faire ?

- Nous devons expliquer à Postfix que pour la remise locale, il doit utiliser cyrus, sinon, ce sera difficile d'envoyer des messages à notre test. Avec uw-imap, c'était simple, la configuration de Postfix par défaut fonctionne, elle utilise procmail pour la livraison locale. Ici, procmail ne fonctionnera pas, il faudra utiliser à la place l'agent proposé par cyrus. Pour l'instant, il n'y a pas grand chose à faire, Postfix connaît l'existence de cyrus par défaut. Il suffit de modifier comme suit dans /etc/postfix/main.cf :

```
mailbox_transport = cyrus
#mailbox_command = procmail -a "$EXTENSION"
```

Spécifier le transport pour mailbox et supprimer l'appel de la commande procmail.

- Le fichier master.cf de Postfix doit contenir une information concernant l'utilisation de cyrus. Cette information est parfois disponible et parfois non, suivant la distribution. Il faut s'assurer que master.cf contient des lignes de ce genre :

```
# The Cyrus deliver program has changed incompatibly, multiple times.
cyrus    unix  -      n      n      -      -      pipe
         flags=R user=cyrus argv=/usr/sbin/cyrdeliver -e -m "${extension}" ${user}
# Cyrus 2.1.5 (Amos Gouaux)
#cyrus   unix  -      n      n      -      -      pipe
         user=cyrus argv=/cyrus/bin/deliver -e -r ${sender} -m ${extension} ${user}
```

Comme vous pouvez le constater, l'agent de livraison de cyrus peut changer de nom et de façon de fonctionner, suivant la version. Il faut donc faire bien attention à ce détail pour que la livraison locale fonctionne correctement avec cyrus. Faites le bon choix en fonction de la version que vous utilisez.

A titre d'exemple, voici le master.cf de la configuration utilisée pour rédiger cet article :

```
# =====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (yes)   (never) (50)
# =====
smtp      inet  n       -       y       -       -       smtpd
#628      inet  n       -       n       -       -       qmqpd
pickup   fifo  n       -       y       60      1       pickup
cleanup  unix  n       -       y       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
#qmgr     fifo  n       -       n       300     1       nqmgr
rewrite  unix  -       -       y       -       -       trivial-rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
flush     unix  n       -       y       1000?   0       flush
smtp      unix  -       -       y       -       -       smtp
showq     unix  n       -       y       -       -       showq
error     unix  -       -       y       -       -       error
local     unix  -       n       n       -       -       local
virtual   unix  -       n       y       -       -       virtual
lmtp      unix  -       -       y       -       -       lmtp
#
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
# The Cyrus deliver program has changed incompatibly.
#
cyrus     unix  -       n       n       -       -       pipe
         flags=R user=cyrus argv=/usr/sbin/cyrdeliver -e -m ${extension} ${user}
uucp      unix  -       n       n       -       -       pipe
         flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
```

```

ifmail    unix    -    n    n    -    -    pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix    -    n    n    -    -    pipe
  flags=Fq user=bsmtp argv=/usr/lib/bsmtp/bsmtp -d -t$nexthop -f$sender $recipient
scalemail-backend unix -    n    n    -    2    pipe
  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store ${nexthop} ${user}
${extension}

# only used by postfix-tls
#smtps    inet    n    -    n    -    -    smtpd
  -o smtpd_tls_wrappermode=yes -o #smtpd_sasl_auth_enable=yes
#587      inet    n    -    n    -    -    smtpd
  -o smtpd_enforce_tls=yes -o smtpd_sasl_auth_enable=no
proxymap  unix    -    -    n    -    -    proxymap
trace     unix    -    -    -    -    0    bounce
verify    unix    -    -    -    -    1    verify

```

Il s'agit ici de la version 2.1.17 de cyrus et de la version 2.1.5 de Postfix

- Enfin, il faut créer un compte sasl pour `essai0`, sinon il aura du mal à lire sa BAL :

```

# saslpasswd2 -c essai0
Password:
Again (for verification):

# sasldblistusers2
essai0@cyclope: userPassword
cyrus@cyclope: userPassword

# testsaslauthd -u essai0 -p essai0
0: OK "Success."
cyclope:/var/spool/cyrus/mail/e/user/essai0#

```

## Tests

- On envoie un message à `essai0` :

```

# mail essai0@cyclope.maison.mrs
Subject: hello world
coucou
.
Cc:
#

```

- On vérifie que Postfix a fait son travail en regardant `/var/log/mail.info` :

```

Nov 15 17:45:40 cyclope postfix/pickup[1072]:
  AA9B1F95: uid=0 from=<root>
Nov 15 17:45:40 cyclope postfix/cleanup[1245]:
  AA9B1F95: message-id=<20031115164540.AA9B1F95@cyclope.maison.mrs>
Nov 15 17:45:40 cyclope postfix/qmgr[491]:
  AA9B1F95: from=<root@cyclope.maison.mrs>, size=315, nrcpt=1 (queue
active)
Nov 15 17:45:41 cyclope postfix/pipe[1249]:
  AA9B1F95: to=<essai0@cyclope.maison.mrs>, relay=cyrus, delay=1,
  status=sent (cyclope.maison.mrs)

```

C'est passé.

- Allons voir dans `/var/spool/cyrus/mail/e/user/essai0` :

```

cyclope:/var/spool/cyrus/mail/e/user/essai0# ls -l
total 16
-rw-----  1 cyrus  mail      577 Nov 15 17:45 1.
-rw-----  1 cyrus  mail      584 Nov 15 17:45 cyrus.cache
-rw-----  1 cyrus  mail      152 Nov 15 16:27 cyrus.header

```



```
-rw----- 1 cyrus mail 128 Nov 15 17:45 cyrus.index
cyclope:/var/spool/cyrus/mail/e/user/essai0#
```

Il y a un nouveau fichier. serait-ce le message ?

```
# cat 1.
Return-Path: <cyrus@cyclope.maison.mrs>
Received: from cyclope.maison.mrs ([unix socket])
  by cyclope.maison.mrs (Cyrus v2.1.14-IPv6-Debian-2.1.14-1)
  with LMTP; Sat, 15 Nov 2003 17:45:41 +0100
X-Sieve: CMU Sieve 2.2
Return-Path: <root@cyclope.maison.mrs>
Received: by cyclope.maison.mrs (Postfix, from userid 0)
  id AA9B1F95; Sat, 15 Nov 2003 17:45:40 +0100 (CET)
To: essai0@cyclope.maison.mrs
Subject: hello world
Message-Id: <20031115164540.AA9B1F95@cyclope.maison.mrs>
Date: Sat, 15 Nov 2003 17:45:40 +0100 (CET)
From: root@cyclope.maison.mrs (root)

coucou

#
```

Oui, c'est le message. Jusque là, tout fonctionne.

- **essai0** pourra-t-il lire ce message par IMAP ?

```
# telnet cyclope.maison.mrs 143
Trying 192.168.0.16...
Connected to cyclope.maison.mrs.
Escape character is '^'.
* OK cyclope.maison.mrs Cyrus IMAP4 v2.1.14-IPv6-Debian-2.1.14-1 server ready
001 login essai0 essai0
001 OK User logged in

# Bien ! déjà l'identification fonctionne

002 select INBOX
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen *)]
* 1 EXISTS
* 1 RECENT
* OK [UNSEEN 1]
* OK [UIDVALIDITY 1068910075]
* OK [UIDNEXT 2]
002 OK [READ-WRITE] Completed

# Bon signe, il existe bien un message dans la BAL

003 fetch 1 BODY[TEXT]
* 1 FETCH (FLAGS (\Recent \Seen) BODY[TEXT] {10})
coucou
)
003 OK Completed

# pas la peine de vérifier dans le détail de l'en-tête, c'est bien sûr le bon message

004 logout
* BYE LOGOUT received
004 OK Completed
Connection closed by foreign host.
cyclope:/var/spool/cyrus/mail/e/user/essai0#
```

Et voilà le travail. Nous savons maintenant configurer :

- **saslauth** pour l'authentification des utilisateurs de la messagerie,

- cyrus pour qu'il utilise les services de saslauth,
- Postfix pour qu'il utilise cyrus comme transport local,

et nous savons utiliser :

- cyrus pour créer des boîtes aux lettres avec cyradm,
- sasl pour créer des authentifications avec saslpasswd2.

### **Juste un détail...**

Nous avons vu en créant une boîte aux lettres avec cyradm que le point (.) était utilisé comme séparateur hiérarchique dans l'arborescence des dossiers...

Comment faire alors, si l'on souhaite créer des utilisateurs qui ont un point dans le nom ?

On ne peut pas. Sauf si l'on ajoute dans imapd.conf la ligne :

```
unixhierarchysep: yes
```

Dans ce cas, le séparateur devient un "slash" (/) et le point devient utilisable dans les noms.

Nous devons écrire alors :

```
cyclope> createmailbox user/essai0  
cyclope> lm  
user/essai0 (\HasNoChildren)  
cyclope>
```

Vous préférerez probablement utiliser cette possibilité.

## Plus loin avec Cyrus

Cyrus sait faire plus de choses qu'un simple accès IMAP à une boîte aux lettres.

Nous savons déjà que Cyrus :

- utilise le format "Maildir", plus souple et plus sûr que "Mailbox",
- peut utiliser divers moyens pour l'authentification des utilisateurs, dont SASL, ce qui évite de devoir créer des comptes d'utilisateurs UNIX juste pour une boîte aux lettres,
- permet de gérer une arborescence de répertoires multi niveaux,
- permet, depuis les versions 2.x, de gérer des noms d'utilisateurs contenant des points "." en remplaçant le séparateur de hiérarchie d'origine (le point, justement) par le séparateur standard d'UNIX (le "slash" /).

Nous ne l'avons pas vu et ne le verrons pas ici, mais cyrus sait aussi :

- placer des filtres pour rejeter des messages considérés comme indésirables,
- permettre l'accès aux BAL des utilisateurs de façon très souple, lorsqu'il y a plusieurs serveurs dans le même domaine.  
Dans un tel cas, il doit être possible d'accéder depuis n'importe quel serveur à une BAL située sur n'importe quel autre serveur. Je n'ai pas testé cette possibilité. J'ai beau avoir des moyens, il ne faut tout de même pas exagérer.

Cyrus est aussi capable de proposer une structure permettant d'exposer des groupes usenet (nntp).

Enfin, et ça, nous allons le tester, Cyrus sait proposer des dossiers partagés.

## Les dossiers partagés

Un dossier partagé peut rendre bien des services dans un intranet, voire sur l'internet.

Il s'agit d'un dossier en tous points comparable à une boîte aux lettres, à part que ce dossier sera au moins visible en lecture par l'ensemble des usagers du serveur. Pour que ce soit intéressant, il faut qu'au moins un usager puisse y déposer des messages. Nous allons voir ce que l'on peut faire avec ça.

### Création d'un dossier partagé

Un dossier partagé va apparaître à la racine de l'arborescence de Cyrus. Alors que les dossiers "Inbox" des utilisateurs sont créés dans cyradm avec la commande :

```
cm user/<nom de l'utilisateur>
```

un dossier partagé sera créé avec la commande :

```
cm <nom du dossier partagé>
```

Pour réaliser cette manipulation, nous revenons à cyclope.maison.mrs, avec quelques comptes d'utilisateurs déjà créés :

```
cyclope:~# cyradm --user cyrus localhost
IMAP Password:

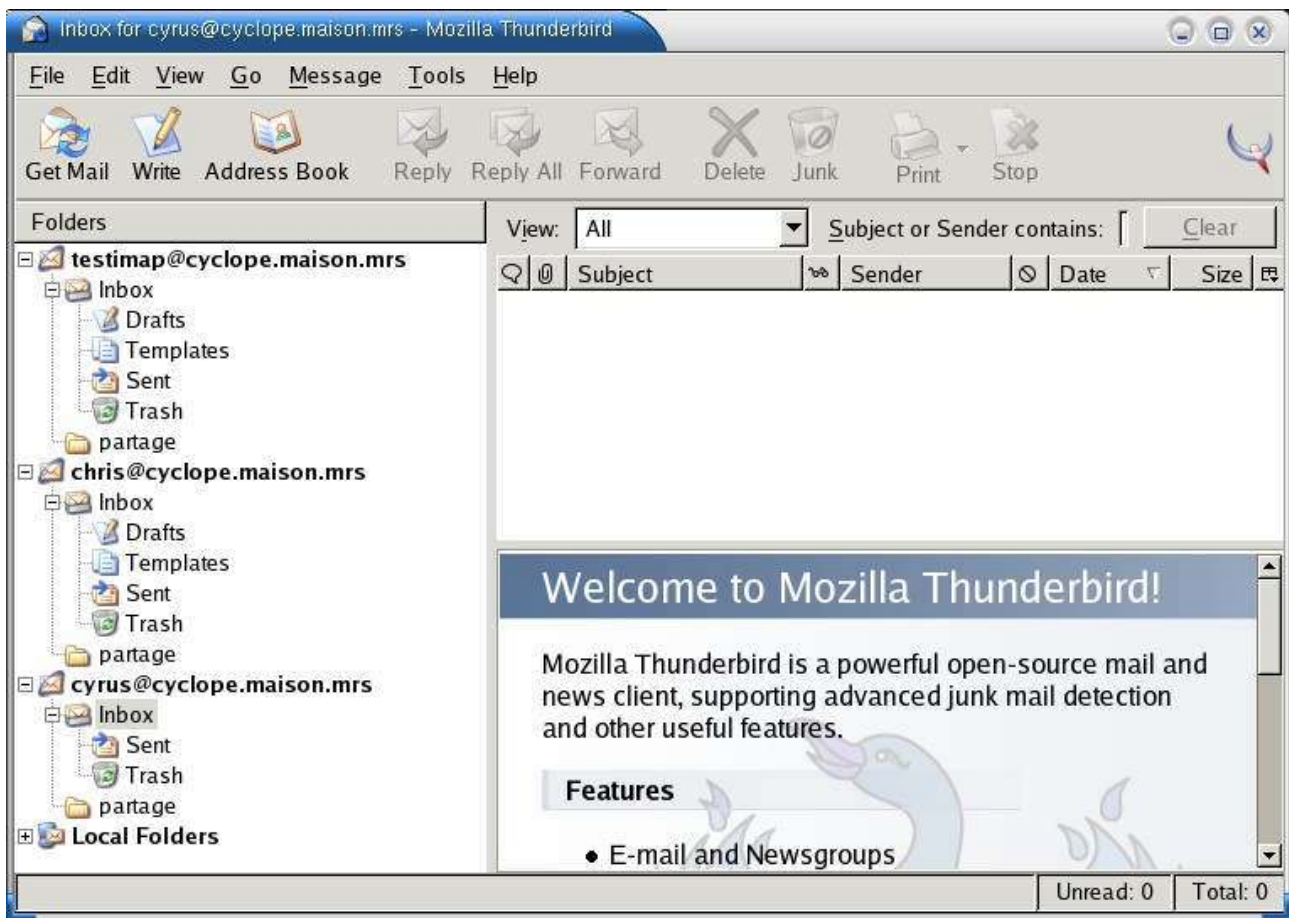
cyclope> cm partage

cyclope> lm
INBOX (\HasChildren)
INBOX/Sent (\HasNoChildren)
INBOX/Trash (\HasNoChildren)
partage (\HasChildren)
user/chris (\HasChildren)
user/chris/Drafts (\HasNoChildren)
user/chris/Sent (\HasNoChildren)
user/chris/Templates (\HasNoChildren)
user/chris/Trash (\HasNoChildren)
user/testimap (\HasChildren)
user/testimap/Drafts (\HasNoChildren)
user/testimap/Sent (\HasNoChildren)
user/testimap/Templates (\HasNoChildren)
user/testimap/Trash (\HasNoChildren)
cyclope>
```

Nous retrouvons :

- testimap, le compte utilisé par les tests précédents,
- chris, un nouveau compte d'utilisateur,
- partage, le répertoire partagé que l'on vient de créer,
- INBOX, qui est le compte de celui qui manipule cyradm, à savoir dans notre cas, l'administrateur de cyrus.

Si, dans Thunderbird, nous créons ces trois comptes et que nous souscrivons à tous les dossiers accessibles pour chaque compte, nous obtenons ceci :



Le répertoire "partage" est accessible depuis tous les comptes, et se situe au même niveau que les "Inbox". Mais que pouvons-nous faire avec ce dossier partagé ?

Voyons les droits avec cyradm :

```
cyclope> lam partage
anyone lrs
cyclope>
```

Il faudrait peut-être regarder de plus près la nomenclature des droits dans la documentation...

- l** lookup    The user may see that the mailbox exists.
- r** read      The user may read the mailbox. The user may select the mailbox, fetch data, perform searches, and copy messages from the mailbox.
- s** seen      Keep per-user seen state. The "Seen" and "Recent" flags are preserved for the user.
- w** write     The user may modify flags and keywords other than "Seen" and "Deleted" (which are controlled by other sets of rights).
- i** insert    The user may insert new messages into the mailbox.

- p** `post` The user may send mail to the submission address for the mailbox. This right differs from the "i" right in that the delivery system inserts trace information into submitted messages.
- c** `create` The user may create new sub-mailboxes of the mailbox, or delete or rename the current mailbox.
- d** `delete` The user may store the "Deleted" flag, and perform expunges.
- a** `administer` The user may change the ACL on the mailbox.

Donc, dans le cas présent, tout le monde (anyone) peut :

- l : voir que ce répertoire existe,
- r : accéder au contenu en lecture et copier les messages dans d'autres dossiers,
- s : les flags "seen" et "recent" sont préservés pour l'utilisateur.

Donc, personne ne peut écrire dedans ?

Non. Ce n'est pas réellement une boîte aux lettres, on ne peut donc y poster un message. Comme personne n'a le droit de copier un message dedans, ça veut dire que ça ne sert à rien.

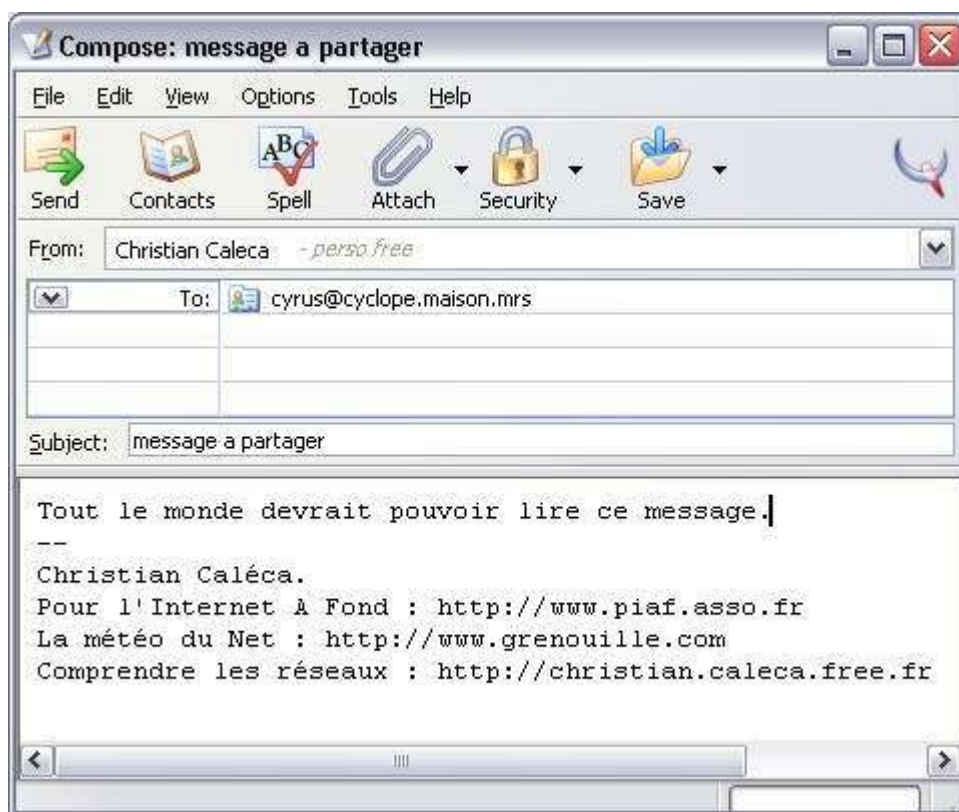
Nous allons maintenant donner à l'utilisateur Cyrus tous les droits sur ce dossier :

```
cyclope> lam partage
anyone lrs
cyclope> sam partage cyrus lrswipcda

cyclope> lam partage
cyrus lrswipcda
anyone lrs
cyclope>
```

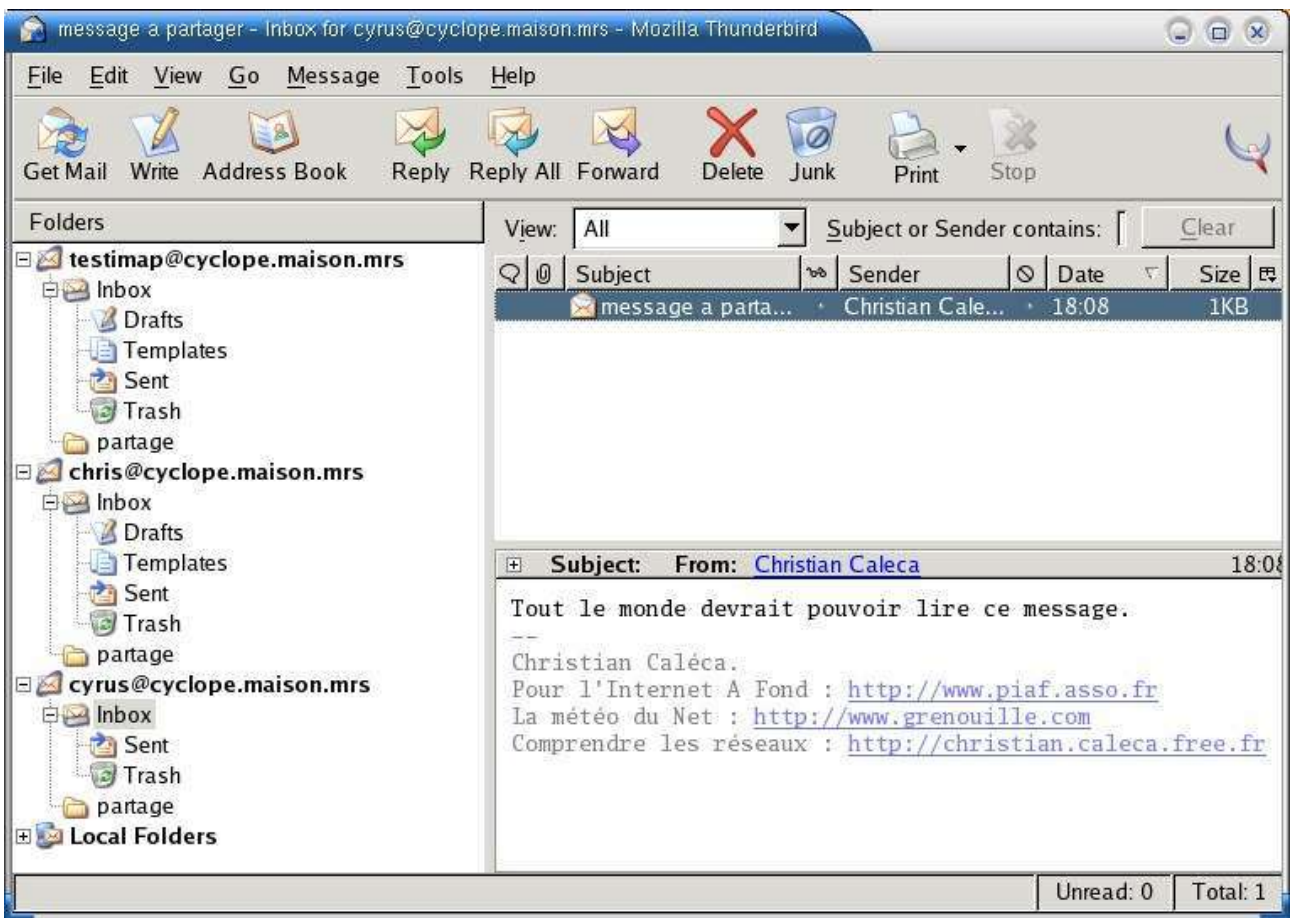
Maintenant que l'utilisateur cyrus a tous les droits sur ce répertoire, il va pouvoir copier des messages dedans.

Nous envoyons un message à [cyrus@cyclope.maison.mrs](mailto:cyrus@cyclope.maison.mrs) :

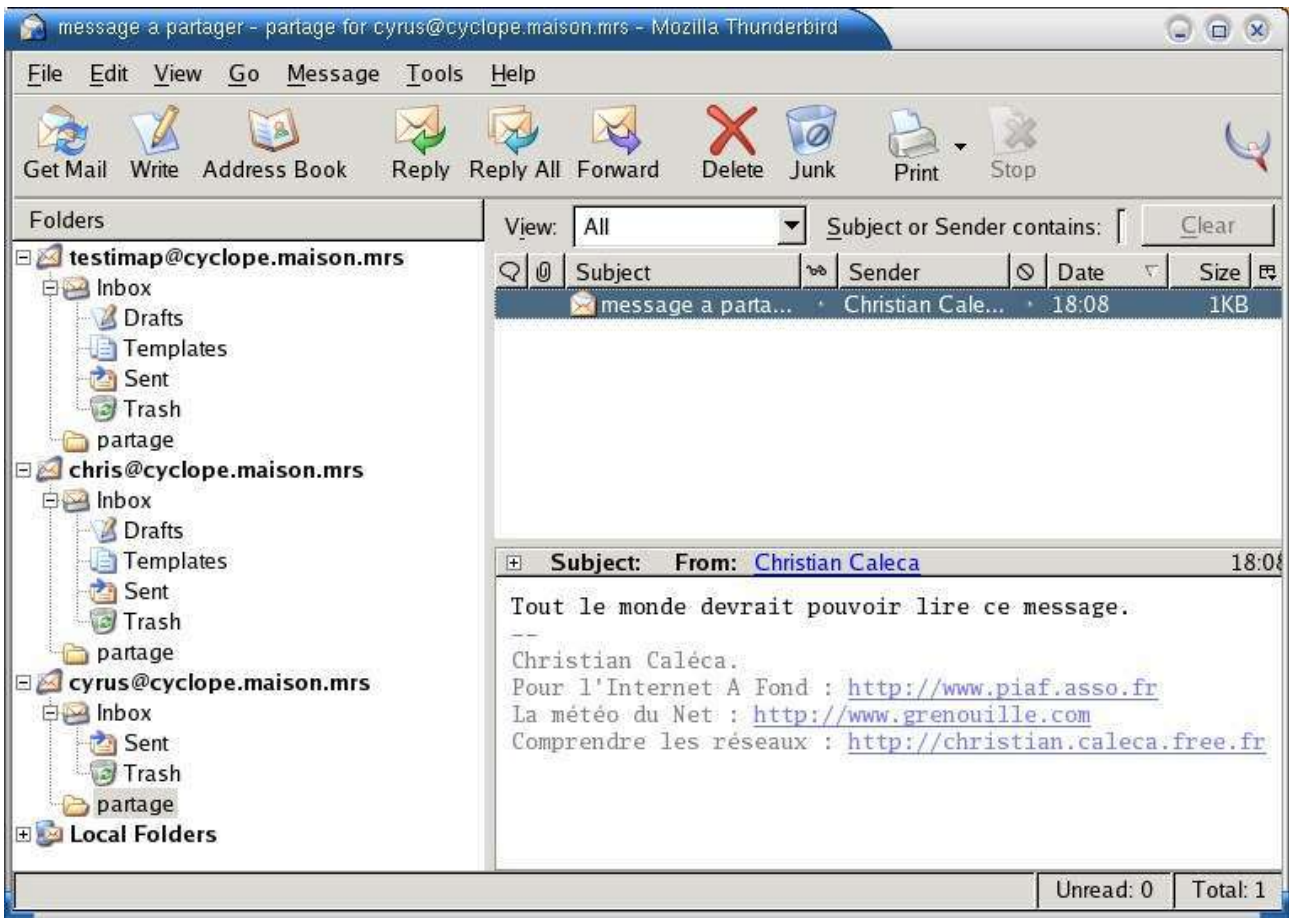


il le reçoit :

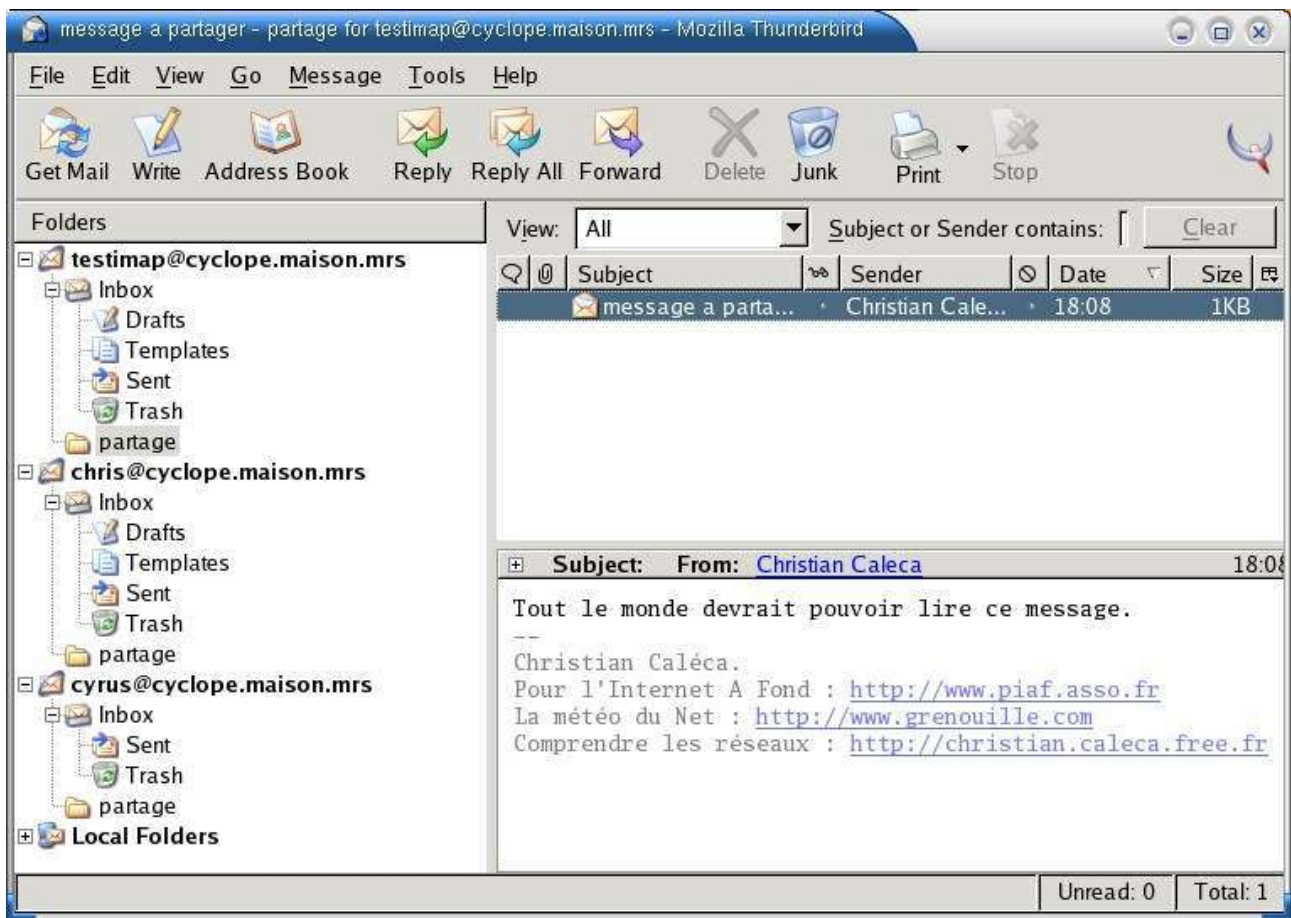




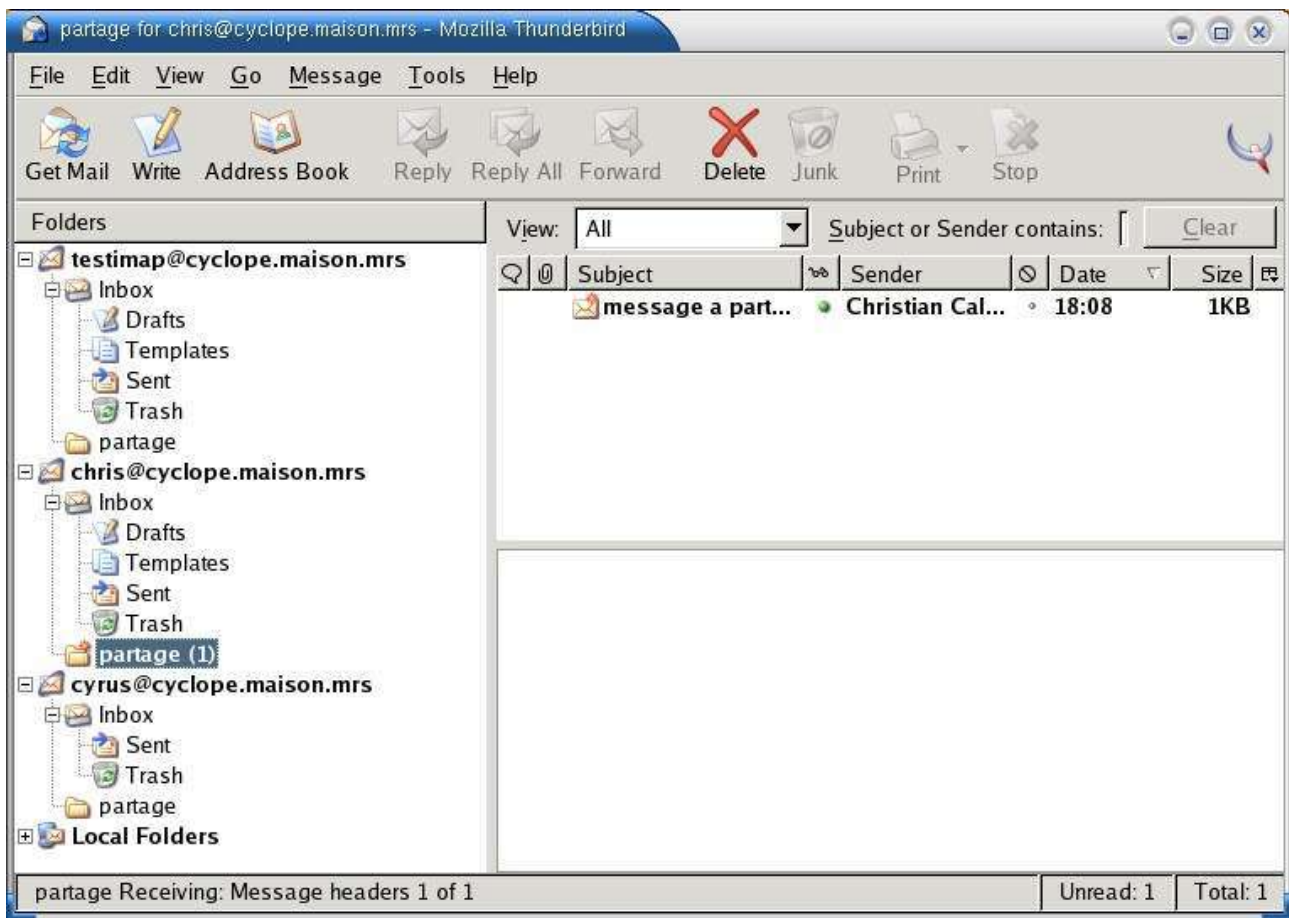
Il le déplace alors dans partage par un "glisser-déposer" :



Et tout le monde pourra en profiter :

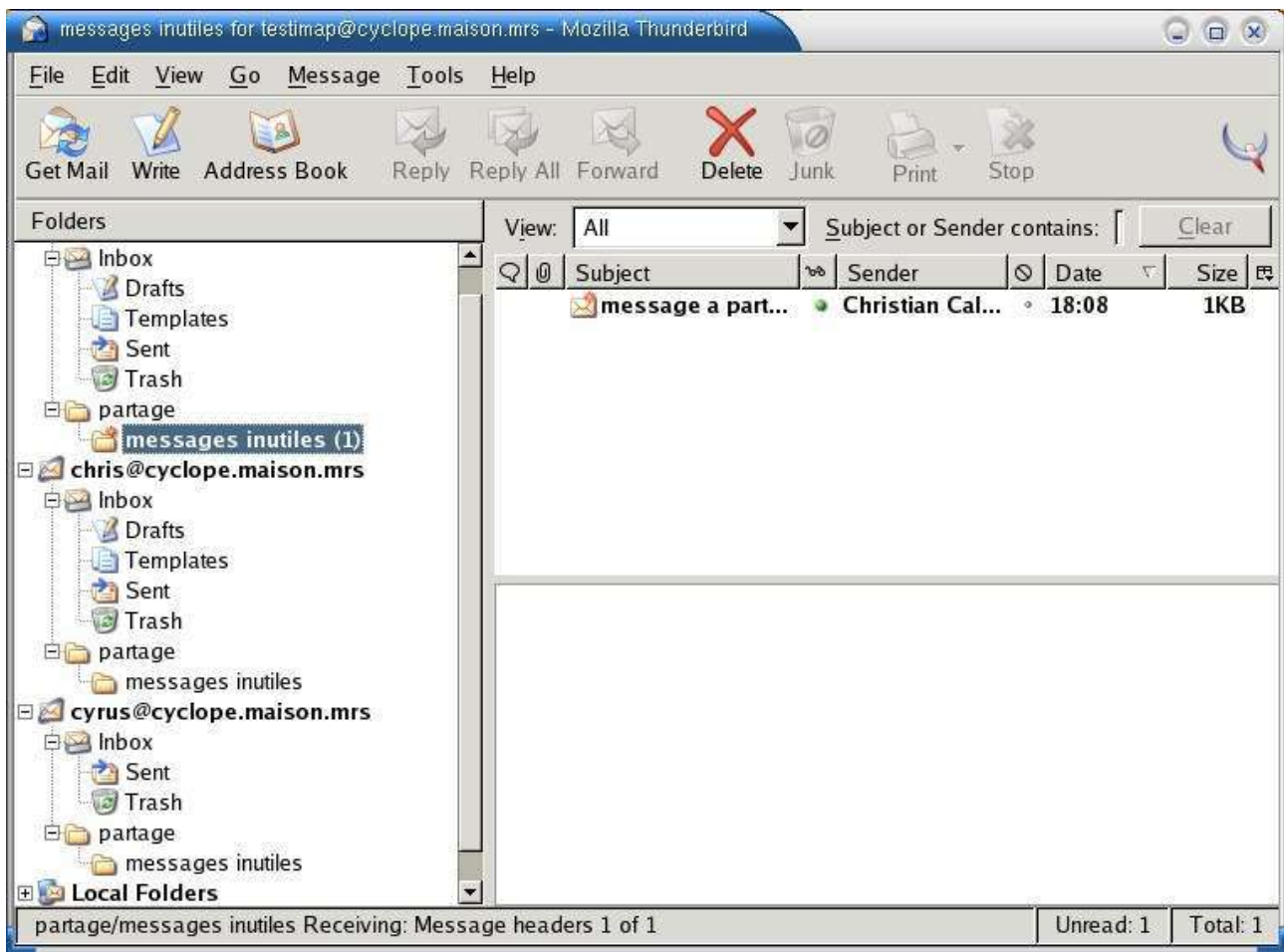


Mieux, ce n'est pas parce que testimap l'aura lu, qu'il apparaîtra comme lu pour tous les autres :



Bien entendu, testimap comme chris pourront copier (et non déplacer) ce message ailleurs, mais ne pourront pas le détruire. Seul cyrus le peut. Cyrus peut également, puisqu'il a tous les droits, créer des sous dossiers, les détruire, supprimer le message qu'il y a mis, ou le déplacer dans un sous-dossier...

Et tous ceux qui auront souscrit aux modifications de structure la verront :



Je vous laisse imaginer les multiples possibilités de ces dossiers, en jouant sur les droits en fonction des utilisateurs. Cette technique peut être très utile pour faire circuler de l'information entre plusieurs personnes.

## Conclusions

Nous n'avons pas vu tout ce qu'il serait possible de voir, mais vous l'avez compris, IMAP est incomparablement plus puissant que POP3, intéressant surtout pour :

- ceux qui lisent leur messagerie depuis plusieurs machines,
- ceux qui ont une connexion plutôt lente.

Il est nécessaire de changer un peu ses habitudes pour exploiter au mieux ce protocole, et de choisir un outil de messagerie qui gère bien IMAP (Outlook Express n'en est pas un).

De nombreux fournisseurs de services proposent désormais ce moyen d'accès à la messagerie. Par exemple : Free, laposte.net, les sites hébergés par OVH, et probablement beaucoup d'autres, mais toujours pas Wanadoo, à l'heure où ce chapitre est rédigé.