sendmail

Pierre David

Laboratoire MASI

Université de Versailles – St Quentin

`Pierre.David@masi.ibp.fr`

`Pierre.David@masi.uvsq.fr`

sendmail

Programme `sendmail`

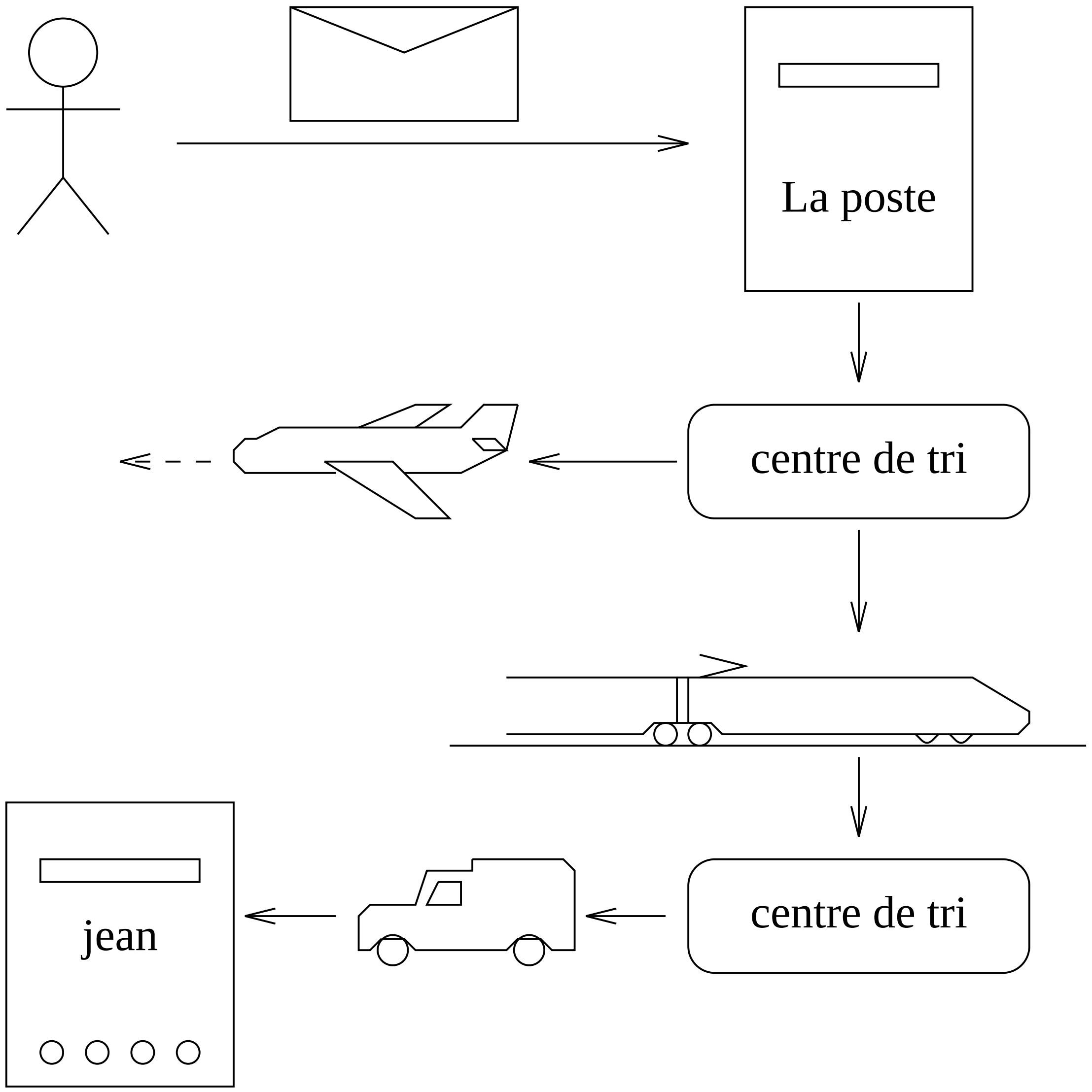
- conçu en 1982 par Eric Allman
- *le* routeur de courrier sur les systèmes dérivés de Berkeley
- souple, puissant
- abscons

sendmail

1. Présentation
2. Rôle de `sendmail`
   (a) interface
   (b) aliases
3. Fichier de configuration
   (a) structure générale
   (b) macros et classes
   (c) *mailers*
   (d) options
   (e) divers
   (f) règles de réécriture
4. Tests
5. Langage *ease*
6. Conclusion

Présentation

Métaphore dans le monde réel :



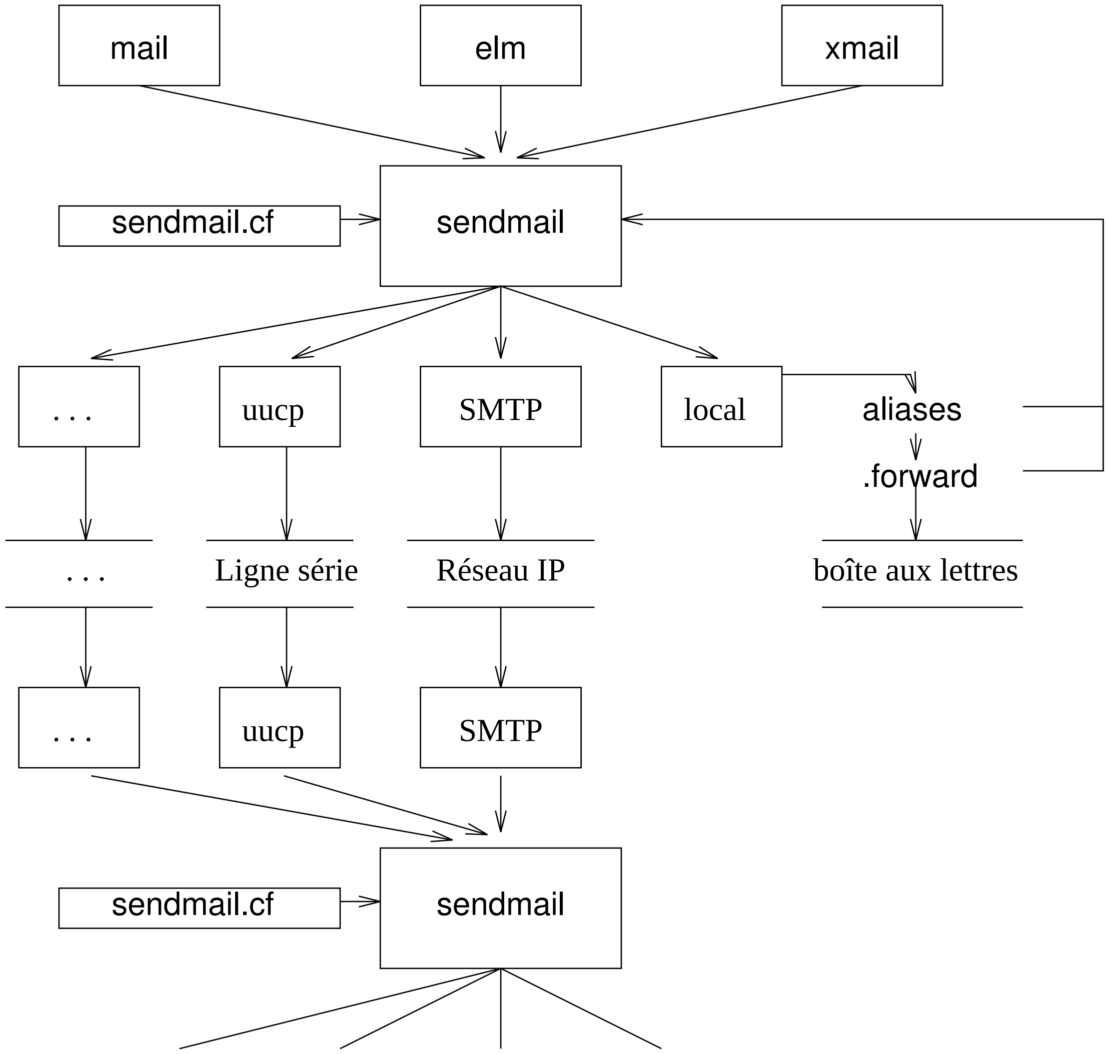
Présentation

Terminologie :

- agent d'usager (*user agent*)
  (`mail`, `elm`, etc.)
- agent de transfert de messages (*message transfert agent*)
  `sendmail`
- agent de transport de messages (*mailer*)
  SMTP, UUCP, etc.
- remise (*local delivery*)
  `mail`
- mémoire de messages (*mailbox*)
  `/usr/spool/mail/jean`

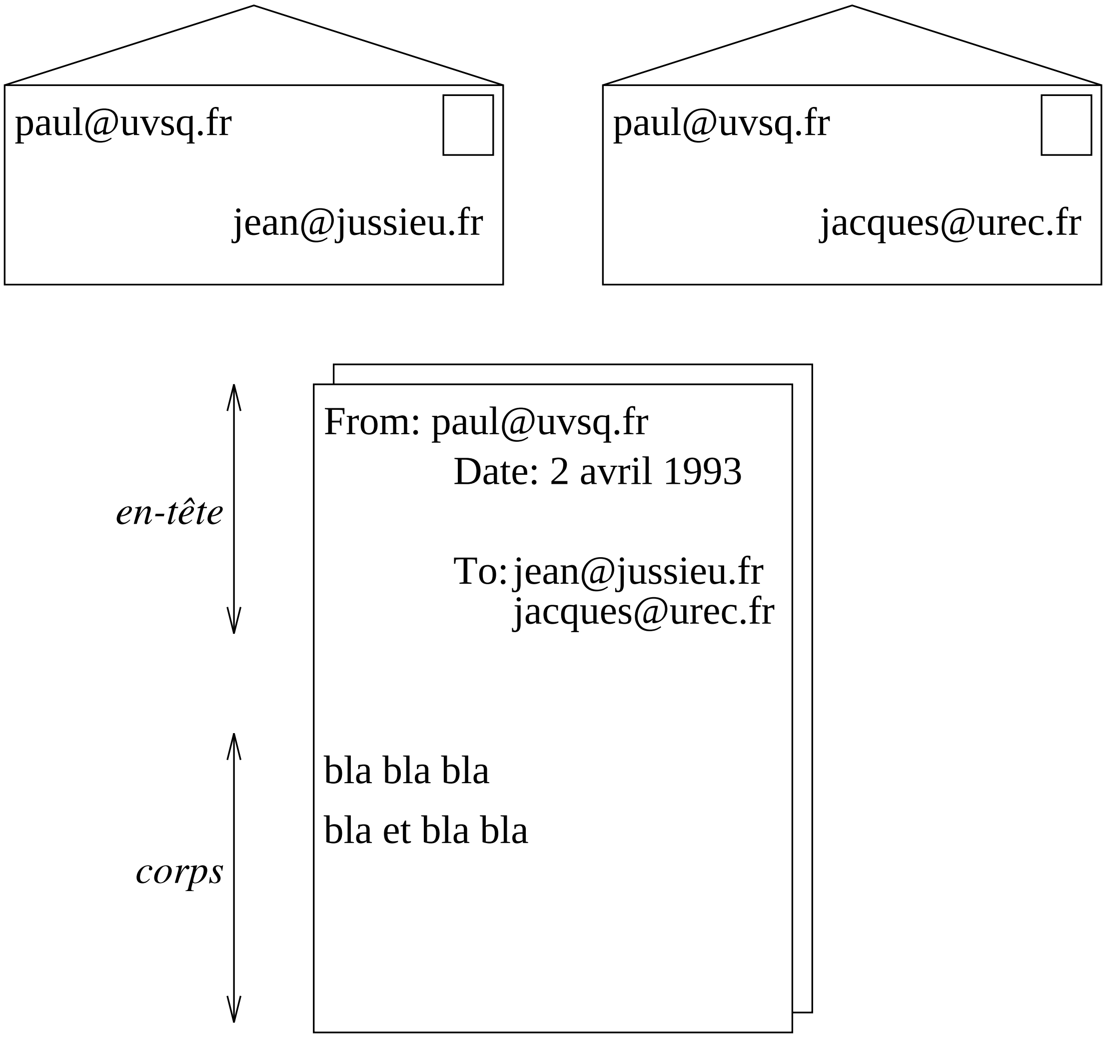
Présentation

Rôle de `sendmail` :



Présentation

Analogie :



Présentation

Lectures recommandées :

RFC 822	format des messages
RFC 821	protocole SMTP
RFC 974	courrier et DNS
RFC 1035	DNS
RFC 1123	prérequis pour les sites Internet

... mais lectures difficiles !

Présentation

La RFC 822 spécifie le format des messages :

- structure d'un message
- format des lignes d'en-tête
- format des dates
- format des adresses

La RFC 822 spécifie également l'utilisateur **Postmaster**

$\Rightarrow$ **tout site doit obligatoirement avoir une boîte aux lettres pour Postmaster (lue régulièrement)**

Présentation – RFC 822

Un message est composé :

- d’une en-tête
- d’un corps

séparés par une ligne (au moins) vide.

Présentation – RFC 822

Une ligne d'en-tête est composée :

- d'un mot-clef
- d'un caractère "`:`"
- d'arguments, éventuellement sur plusieurs lignes

Exemple :

```
Return-Path: <paul@uvsq.fr>
Received: from soleil.uvsq.fr
          by shiva.jussieu.fr (1.0/j3.1)
          at Fri, 02 Apr 1993 09:30:45 +0100
From: Paul Ochon <paul@uvsq.fr>
Subject: essai
To: jean@jussieu.fr (Jean Breille),
    "Jacques Selere" <jacques @ urec.fr>
Date: Fri, 02 Apr 1993 09:30:15 +0100
```

Présentation – RFC 822

`Return-Path`	Ajouté à la fin seulement
`Received`	Vérification
`From`	Émetteur
`Sender`	Émetteur réel
`Reply-To`	Adresse de retour
`Date`	Date d'émission
`To`	Destinataires
`Cc`	Carbon copy
`Bcc`	Blind carbon copy
`Message-Id`	Identificateur unique
`In-Reply-To`	Référence au message original
`References`	Références aux messages précédents
`Keywords`	Mots-clefs
`Subject`	Sujet du message
`Comments`	Commentaires
`Encrypted`	Message crypté et méthode utilisée
`X-???`	Messages d'extension

Présentation – RFC 822

Après un re-routage des courriers :

```
Resent-From
Resent-Sender
Resent-Reply-To
Resent-To
Resent-Cc
Resent-Date
Resent-Bcc
Resent-Message-Id
```

Présentation – RFC 822

Format des dates :

- jour de la semaine optionnel
- quantième (2 chiffres)
- mois (3 lettres)
- année (2 ou 4 chiffres)
- heure
- fuseau horaire

Exemple :

```
Fri, 02 Apr 1993 09:30:15 +0100
```

Présentation – RFC 822

Format des adresses :

- adresses globales

```
jean @ jussieu.fr
jean (Jean Breille) @ jussieu.fr
```

- adresses avec routage explicite

```
<@urec.fr,@uvsq.fr:jean@jussieu.fr>
<jean@jussieu.fr>
Jean Breille <jean@jussieu.fr>
"Jean Breille" <jean@jussieu.fr>
```

  Note : le routage explicite est déconseillé

- adresses littérales

```
jean @ [134.157.0.129]
```

  Note : les adresses littérales sont déconseillées

- extension RFC 1123 : routage avec `%`

```
jean % jussieu.fr @ uvsq.fr
```

Commentaires : entre parenthèses

Présentation – RFC 821

220 shiva.jussieu.fr Sendmail 1.0 ready at Fri, 2 Apr 93 09:30:45 GMT

HELO soleil.uvsq.fr

250 shiva.jussieu.fr Hello soleil.uvsq.fr, pleased to meet you

MAIL From: <paul@uvsq.fr>

250 <paul@uvsq.fr>... Sender ok

RCPT To: <jean@jussieu.fr>

250 <jean@jussieu.fr>... Recipient ok

DATA

354 Enter mail, end with "." on a line by itself

From: paul@uvsq.fr (Paul Ochon)

Subject: essai

To: jean@jussieu.fr (Jean Breille),

"Jacques Selere" <jacques @ urec.fr>

Date: Fri, 02 Apr 1993 09:30:15 +0100

ceci est un essai

.

250 Mail accepted

QUIT

Présentation – RFC 821

Limitations (minimum admissible) :

- 7 bits
- nom d'utilisateur < 64 caractères
- nom de domaine < 64 caractères
- nombre de destinataires < 100
- lignes < 1000 caractères

⇒ Extended SMTP (RFC 1425)

Présentation – RFC 974

Resource records de type MX :

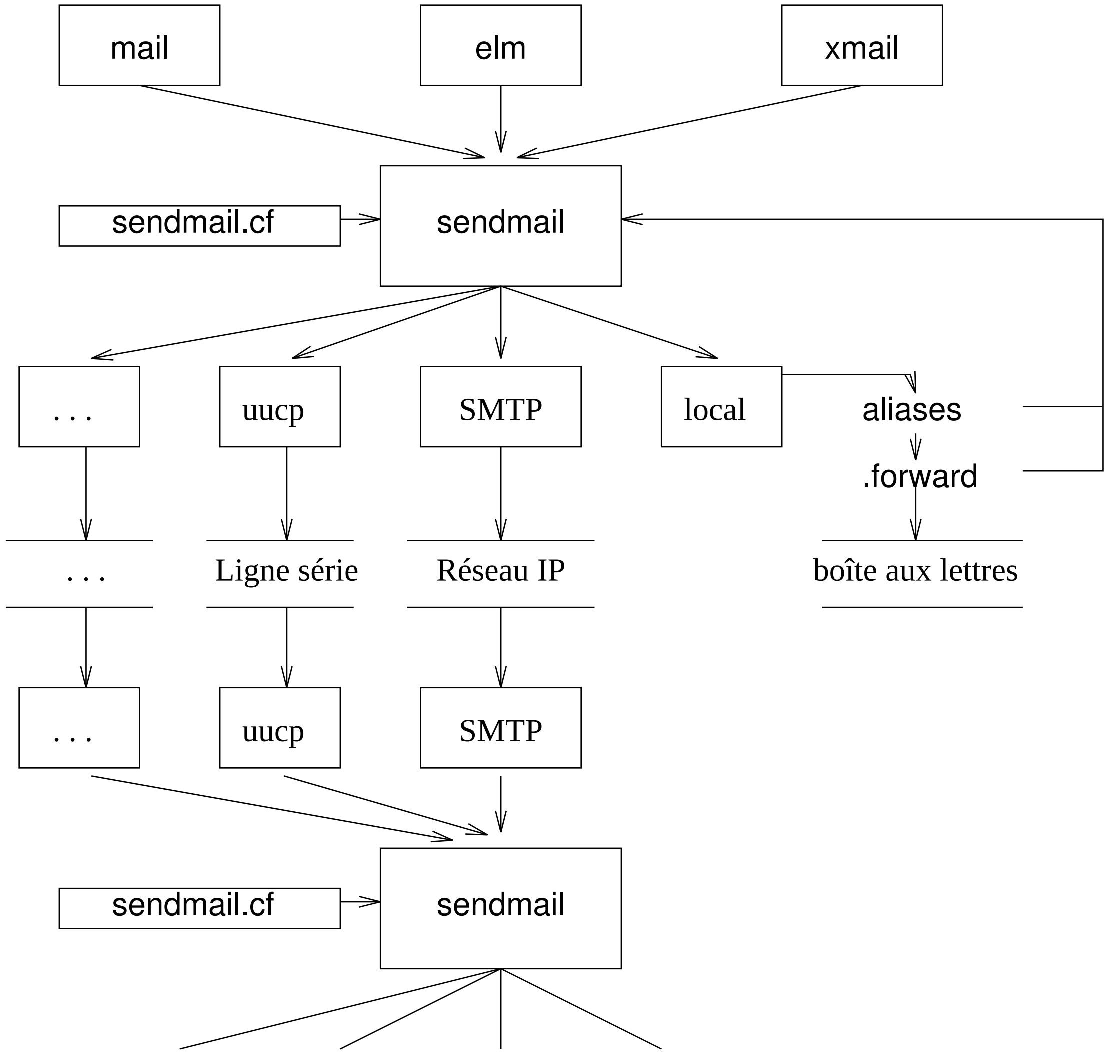
Exemple :

```
uvsq.fr IN MX 100 soleil.uvsq.fr.
uvsq.fr IN MX 200 shiva.jussieu.fr.
```

Traitement :

- s'il n'y a pas de MX, alors envoyer le courrier directement à la machine
- retirer le MX pointant sur la machine locale, ainsi que tous les MX moins prioritaires
- essayer les MX par ordre de priorité

Fonctions



mail
elm
xmail
sendmail.cf
sendmail
. . .
uucp
SMTP
local
aliases
.forward
. . .
Ligne série
Réseau IP
boîte aux lettres
. . .
uucp
SMTP
sendmail.cf
sendmail

Fonctions

Appel de `sendmail` :

- par un agent d'usager :
- directement par l'utilisateur :

  $ **/usr/lib/sendmail jean@jussieu.fr jacques@urec.fr**

  From: paul

  To: jean@jussieu.fr, jacques@urec.fr

  Subject: essai

  Ceci est un essai

  ^ D

  $
- par un agent de transport de messages (ex : uucp)
- par un autre `sendmail` : connection SMTP

Fonctions

Qui est qui ?

- répertoire `/usr/lib/` :

`sendmail`	le programme
`sendmail.cf`	fichier de configuration
`sendmail.fc`	configuration “compilée”
`sendmail.hf`	aide lors du dialogue SMTP
`sendmail.st`	informations comptables
`aliases`	les aliases en clair
`aliases.pag`	version “compilée” des aliases
`aliases.dir`	version “compilée” des aliases

- répertoire `/usr/spool/mqueue/` :
  la file d’attente
- répertoire `/usr/spool/mail/` :
  les boîtes aux lettres

Note : les chemins peuvent être modifiés.

Fonctions

Les aliases (définis dans `aliases`) permettent de re-router le courrier local :

- nom complet d'utilisateur :
  `Paul.Ochon: paul`
- utilisateur virtuel :
  `paul: paul@uvsq.fr`
- programme :
  `pg: "| /bin/cat | /bin/sort > /tmp/toto"`
- liste de diffusion :
  ```
  liste: paul@uvsq.fr, jean@jussieu.fr
  owner-liste: paul@uvsq.fr
  liste-request: paul@uvsq.fr
  ```
- liste de diffusion gérée par un utilisateur :
  `liste: :include:/users/paul/liste`

Ne pas oublier d'utiliser `newaliases` !

Pas de nom non local !

Fonctions

Fichiers `.forward` :

- même syntaxe que les aliases
- consulté *après* les aliases

Configuration

Le fichier `sendmail.cf` contient la configuration de `sendmail` :

- règles de réécriture
  $\Rightarrow$ réécriture des adresses
  $\Rightarrow$ sélection du routage du courrier
- définition de *macros*
  $\Rightarrow$ souplesse de configuration
  $\Rightarrow$ accès aux spécificités du système
- définition de *mailers*
  $\Rightarrow$ facilité d'addition de “réseaux”
- diverses options

Configuration – Macros

Macros

- Macros

  DDuvsq.fr

  DMsoleil.$D

  De$j Sendmail $v ready at $b

  Dq$?x $x <$g>$| $g $.

- Classes

  CLlabo1 labo2 labo3

  CLlabo4 labo5

  La classe **w** est l’ensemble des noms du site.

- Classes lues dans des fichiers

  FV/etc/hosts %*[0-9.] %s

Les minuscules sont utilisées par le système. Il faut utiliser les majuscules pour ses propres macros.

Configuration – Macros

Macros standard

a	date d'envoi en format RFC822
b	date courante en format RFC822
c	nombre de “hops”
d	date courante en format Unix
e	message d'introduction SMTP
f	nom de l'expéditeur (“From”)
g	nom de l'expéditeur vu du destinataire
h	host du destinataire
i	numéro dans la file d'attente
j	nom de domaine officiel pour ce site
l	format de la ligne “From”
m	nom de domaine
n	nom du démon pour les messages d'erreur
o	caractères séparateurs
p	numéro de processus de `sendmail`
q	format par défaut des noms d'expéditeur
r	protocole utilisé
s	host de l'expéditeur
t	représentation numérique de la date
u	utilisateur destinataire
v	numéro de version de `sendmail`
w	nom de host de ce site
x	nom complet de l'expéditeur
z	répertoire par défaut du destinataire

Configuration – Mailers

Un *mailer* est défini par :

- un nom (utilisé dans le fichier de configuration)
- un programme et ses paramètres
- des caractères de fin de ligne
- des flags
- une taille maximum de messages
- des règles de réécriture pour les adresses :
  - d'origine
  - de destination

Exemples :

Muucp, P=/usr/bin/uux, F=DFMUshu, M=100000, S=13, R=23, A=uux - $h!rmail ($u)

Mlocal, Path=/bin/mail, Flags=DFMPlms, Sender=10, Recipient=20, Argv=mail -d $u

Configuration – Mailers

Mailers obligatoires :

- **local**

  utilisé pour la remise (*local delivery*)

- **prog**

  utilisé pour les aliases sous forme de programmes

- **error**

  défini implicitement

  utilisé explicitement

Configuration – Mailers

Cas particulier : SMTP

Msmtp, P=[IPC], F=CDFMXmu, S=11, R=21, E=\r\n, A=IPC $h

⇒ pas de programme externe

⇒ SMTP est inclus dans `sendmail`

Configuration – Divers

Définition des champs d'en-tête : ajout du champ s'il n'existe pas déjà.

Exemples :

- **HDate: $a**
- **H?F?From: $q**

Champs supplémentaires gérés par `sendmail`

`Full-Name`	Nom complet
`Return-Receipt-To`	Accusé de remise
`Errors-To`	Destinataire des erreurs
`Apparently-To`	Pas de To:, Cc: ou Bcc:
`Precedence`	Priorité du message

Configuration – Divers

- précédence des messages

  affecte la priorité du traitement des messages

  Pjunk=-100

- *trusted users*

  utilisateurs ayant le droit d'envoyer un courrier pour le compte d'un autre utilisateur

  Tuucp

Configuration – Divers

Options :

- **OT5d** : temps de rémanence dans la queue avant rejet
- **OA/usr/lib/aliases** : fichier des aliases
- **OX20** : charge à partir de laquelle `sendmail` n'accepte plus les connexions SMTP
- **OL10** : niveau de *log*
- etc.

⇒ consulter la documentation...

Configuration – Règles de réécriture

Les règles de réécriture servent à :

- reformuler les adresses :

  exemple : **toto@voisin.uucp** ⇒ **voisin!toto**

- choisir un *mailer* :

  exemples :

  voisin!toto ⇒ UUCP

  jean@jussieu.fr ⇒ SMTP

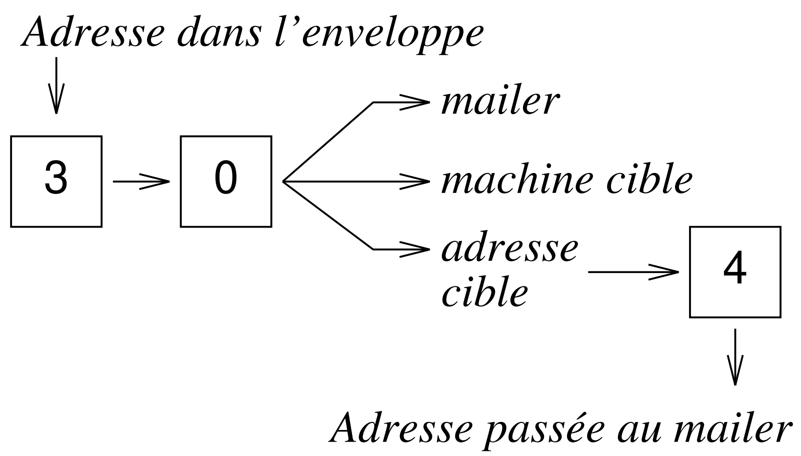
Configuration – Règles de réécriture

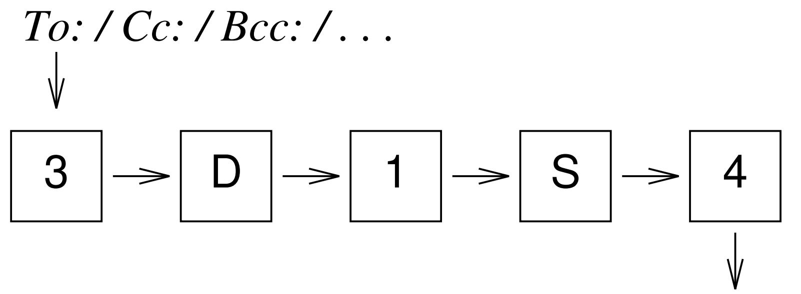
Les règles sont regroupées en ensembles :

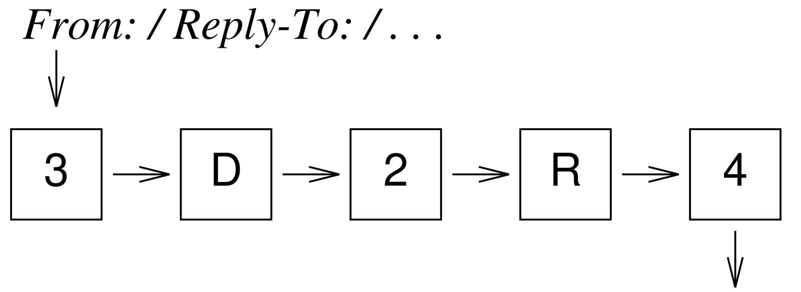
- repérés par un numéro
- accomplissant une transformation sur une adresse

Certaines règles ont des rôles prédéfinis

Configuration – Règles de réécriture







Configuration – Règles de réécriture

Règles avec des rôles prédéfinis :

0	sélection du *mailer*
1	réécriture des adresses d'émetteur
2	réécriture des adresses de destinataire
3	canonisation des adresses
4	dé-canonisation
D	ajout de *@host* si nécessaire (flag C du mailer)
S	règle spécifique au *mailer*
R	règle spécifique au *mailer*

Note : certaines versions de `sendmail` (IDA) utilisent en plus d'autres règles prédéfinies.

Configuration – Règles de réécriture

Canonisation : utilisée pour représenter toutes les adresses possibles avec un format unique.

Plusieurs formats possibles. Le plus traditionnel est :

partie-locale **< @** *machine-cible* **>** *routage*

Configuration – Règles de réécriture

Exemple :

Paul.Ochon@uvsq.fr

Canonisation

Paul.Ochon < @ uvsq.fr >

Partie "distante"
Utilisée par la machine distante

Machine cible
Utilisée pour router le message

Partie "locale" à la machine cible
(nom de l'utilisateur)

Configuration – Règles de réécriture

Exemple :

Paul Ochon <@jussieu.fr : Paul.Ochon@uvsq.fr>

Canonisation

<@jussieu.fr> : Paul.Ochon@uvsq.fr

Partie "distante"
Utilisée par la machine distante

Machine cible
Utilisée pour router le message

Partie "locale"
vide

Configuration – Règles de réécriture

Exemple :

site1 ! site2 ! site3 ! toto

Canonisation

site2 ! site 3 ! toto <@site1 . UUCP>

Partie "distante"
vide

Machine cible
Utilisée pour router le message

Partie "locale"
Adresse passée au mailer

Configuration – Règles de réécriture

Sélection du *mailer* : la règle 0 a pour but de déterminer le *mailer* à appeler

⇒ la règle 0 décide du *routage* du message.

La règle 0 doit sélectionner un triplet :

- nom de *mailer* :
  la sélection d'un *mailer* influe également la réécriture des adresses dans l'en-tête

- nom de site :
  utilisé par le *mailer*
  (ex: nom de machine pour SMTP, nom de site pour UUCP, etc.)

- nom d'utilisateur :
  envoyé au *mailer*
  ⇒ la nouvelle enveloppe

Configuration – Règles de réécriture

Un ensemble de règles est composé de règles (!).

Une règle (une ligne) est composée de champs séparés par des tabulations :

R*motif*   *remplacement*   *commentaires*

- les règles sont appliquées séquentiellement dans l'ensemble
- si une règle correspond avec le *motif*, l'ensemble est transformé selon le *remplacement*
- si le remplacement était possible, la règle est répétée
- si le remplacement n'est pas possible, on passe à la règle suivante.

Configuration – Règles de réécriture

```
# Ensemble 15
# transforme les adresses "..%..%.." en "..%..@.."
# et les adresses "..@..@.." en "..%..@.."
S15
R$+%$+         $1@$2          supprime les %
R$+@$+@$+      $1%$2@$3       conserve le dernier @
```

$a \% b \% c \% d$

R$+%$+ $\downarrow$

$\underline{a} \% \underline{b \% c \% d} \longrightarrow a \,@\, b \% c \% d$

$\underline{a \,@\, b} \% \underline{c \% d} \longrightarrow a \,@\, b \,@\, c \% d$

$\underline{a \,@\, b \,@\, c} \% \underline{d} \longrightarrow a \,@\, b \,@\, c \,@\, d$

$a \,@\, b \,@\, c \,@\, d$

R$+@$+@$+ $\downarrow$

$\underline{a} \,@\, \underline{b} \,@\, \underline{c \,@\, d} \qquad a \% b \,@\, c \,@\, d$

$\underline{a \% b} \,@\, \underline{c} \,@\, \underline{d} \qquad a \% b \% c \,@\, d$

$a \% b \% c \,@\, d$

Configuration – Règles de réécriture

“Mots-clefs” pour le *motif* (partie gauche) :

$*	zéro ou plusiers tokens
$+	un ou plusiers tokens
$–	exactement un token
$*x*	la macro *x*
$=*x*	tout token dans la classe *x*
$˜ *x*	un token absent de la classe *x*

Exemple :

La classe V est la classe des sites voisins

La macro D est mon nom de domaine

R$*<@$=V.$D>$* $1<@LOCAL>$3

Configuration – Règles de réécriture

“Mots-clefs” pour le *remplacement* (partie droite) :

$*n*	le token correspondant dans le *motif*
$>*n*	appeler l’ensemble de règles *n*
$<*prog*	appelle le programme spécifié
$#*m*$@*h*$:*u*	sélection du triplet dans la règle 0
$[*h*$]	canonise un nom de site
$@	applique la règle et *termine* l’ensemble
$:	empêche l’application itérative d’une règle

Configuration – Règles de réécriture

Exemples :

```
CB bitnet earn

DR frmop11.cnusc.fr

R$*<@$*>$*          $:$1<@$[$2$]>$3

R$*                 $:$>9$1

R$-                 $#local $:$1

R$*<@$-.$=B>$*      $#smtp $@$R $:$1<@$2.bitnet>$4

R$*<@$*>$*          $#smtp $@$2 $:$1<@$2>$4

R$*                 $#error $:unrecognizable sequence $1
```

Configuration – Règles de réécriture

Les règles *S* et *R* dépendant des *mailers* permettent le maximum de souplesse. Par exemple :

```
DMmon-domaine.fr
Msmtp, P=[IPC], F=CDFMXmu, S=11, R=21, ... A=IPC $h
Msmtplocal, P=[IPC], F=CDFMXmu, S=12, R=22, ... A=IPC $h

S0
...
R$*<@$*.$M>$*        $#smtplocal $@$2.$M $:$1<@$2.$M>$3
R$*<@$*>$*           $#smtp $@$2 $:$1<@$2>$3

# retrait du nom de machine pour les envois à l'extérieur
S11
R$*<@$*.$M>$*        $1<@$M>$3

# envoi à l'intérieur : on laisse le nom de machine
S12
# rien
```

Configuration – Règles de réécriture

Extensions utiles : réécriture à l'aide d'une table externe.

- `sendmail` de Sun

  `DYaliases.out`

  `R$*<@$=w>$*        $:${Y $1$}<@$2>$3`

  Utilisation des maps NIS

- `sendmail` avec extensions IDA

  `OKN/usr/local/etc/revaliases`

  `R$*<@$=w>$*        $:$(N $1$)<@$2>$3`

  site ftp : `uxc.cso.uiuc.edu` (sendmail 5.65c + IDA 1.4.4.1)

- `sendmail` version 6

  `KN/usr/local/etc/revaliases`

  `R$*<@$=w>$*        $:$(N $1$)<@$2>$3`

  site ftp : `ftp.cs.berkeley.edu` pour les développeurs et testeurs

Tests

Vérifier :

- l'alias de *Postmaster*
- la boîte aux lettres de *Postmaster*
- la sortie de `syslogd`
- qu'il n'y a pas de version "compilée" (`sendmail.fc`)
- (IBM seulement) qu'il n'y a pas de version "compilée" (`sendmail.cfDB*`)

Tests

Pouvez-vous envoyer un courrier local ?

$ **/usr/lib/sendmail -v toto**

From: peu-importe

To: toto

essai

.

toto... Connecting to via local...

toto... Sent

$\Rightarrow$ beaucoup de règles marchent !

Tests

Les autres machines ne peuvent pas envoyer de courrier :

⇒ vérifier que `sendmail` est actif.

$ **telnet localhost smtp**

Connected to localhost.

Escape character is '^]'.

220 shiva.jussieu.fr Sendmail 1.0 ready at Fri, 2 Apr 93

09:30:45 GMT

quit

221 shiva.jussieu.fr closing connection

Connection closed by foreign host.

⇒ `sendmail` est à l'écoute des connexions SMTP

Tests

Test simple :

mail moi%mon-site@site-distant

Autre test :

Tests

Pour vérifier les règles de réécriture :

$ **/usr/lib/sendmail -bt**

ADDRESS TEST MODE

Enter <ruleset> <address>

> **0 pda@masi.ibp.fr**

rewrite: ruleset 3 input: pda @ masi . ibp . fr

rewrite: ruleset 8 input: pda < @ masi . ibp . fr >

rewrite: ruleset 8 returns: pda < @ masi . ibp . fr >

rewrite: ruleset 3 returns: pda < @ masi . ibp . fr >

rewrite: ruleset 0 input: pda < @ masi . ibp . fr >

rewrite: ruleset 9 input: pda < @ masi . ibp . fr >

rewrite: ruleset 9 returns: pda < @ masi . ibp . fr >

rewrite: ruleset 0 returns: $# smtp $@ masi . ibp . fr $: pda < @ masi . ibp . fr >

En cas de sérieux problème, utiliser les options de débug :

$ **/usr/lib/sendmail -bt -d21.99**

⇒ beaucoup d'information...

Tests

Problèmes pour voir ce que fait un *mailer*

$\Rightarrow$ Écrire un nouveau *mailer* :

```
#!/bin/sh
exec >> /tmp/mail.log
echo ----------------
date
echo arguments :
for i ; do echo $i ; done
echo message :
cat
exit 0
```

Remplacer le *mailer* suspect par un autre :

Mlocal, Path=/bin/mail, Flags=DFMPlms, Sender=10, Recipient=20, Argv=mail -d $u

$\Rightarrow$

Mlocal, Path=/tmp/mon-mailer, Flags=DFMPlms, Sender=10, Recipient=20, Argv=mail -d $u

Langage ease

`sendmail` doit être rapide

$\Rightarrow$ syntaxe simple à analyser

$\Rightarrow$ syntaxe absconse

“Mot-clefs” sur une lettre (D, C, M, R, etc.)

Classes et macros sur une lettre

Tokens peu mnémotechniques ($>, $@, etc.)

Solution ? Un langage de plus haut niveau :

- avec des vrais identificateurs
- avec des structures de contrôle

Langage ease

Exemple :

```
field
	zero_ou_plus : match (0*) ;
	un_ou_plus : match (1*) ;
	un_seul : match (1) ;

ruleset conversion_pourcent {
	if (un_ou_plus % un_ou_plus)
		retry ($1@$2) ;
	if (un_ou_plus @ un_ou_plus @ un_ou_plus)
		retry ($1%$2@$3) ;
}
```

Conclusion

`sendmail` est *le* routeur de courrier depuis 1982.

Principe de conception bon $\Rightarrow$ rester plus de 10 ans dans le monde fluctuant de la messagerie naissante.

- 1982 : RFC 822
- 1992 : RFC 1341 – MIME
  Multipurpose Internet Mail Extensions
- 1993 : RFC 1425 – SMTP Service Extensions

La messagerie Internet se perfectionne, `sendmail` est toujours là !