

Mutt : messagerie console



- Objet : mutt La messagerie en ligne de commande.
- Niveau requis : [avisé](#)
- Commentaires : *Expliquer comment installer et configurer Mutt avec des règles de filtres et tout et tout...*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi : [à-compléter](#)
 - Création par [martinux_qc](#) 21/06/2013
 - Testé par [paskal](#) le 05/07/2013
- Commentaires sur le forum : [ici](#)¹⁾

Introduction au courrier électronique

Si vous n'êtes pas un fan des clients de messagerie électronique à interfaces graphiques fantaisistes, ou si vous voulez seulement essayer d'autres clients de messagerie avant de choisir celui qui vous convient le mieux, voici une méthode facile pour commencer à utiliser ces puissants outils en ligne de commande :

- fetchmail,
- procmail,
- mutt,
- smtp,
- urlview,
- postfix,
- msmtpt

Ces programmes ne sont pas seulement puissants et très personnalisables, mais aussi petits et rapides. Lorsque vous serez à l'aise avec ce système de courrier électronique, vous serez étonné par tout ce qu'il vous permet de faire.

Ce tutoriel n'étant qu'un guide de démarrage, nous n'aborderons pas de « Mail Transfert Agent » (MTA ou agent de transfert de courrier) tel que sendmail, postfix ou exim.

En d'autres termes, pas de configuration complexe de MTA.

De cette façon, l'utilisation du **port 25** pour le service de courrier n'est pas nécessaire.

Nous pouvons nous en passer car **fetchmail** peut transférer le courrier qu'il rapatrie directement au « Mail Delivery Agent²⁾ » au lieu de le diriger vers le port 25.

Et nous n'avons pas besoin d'un MTA complexe pour transmettre le courrier sortant au serveur SMTP de notre fournisseur d'accès.

Installation des programmes requis

Voici les programmes dont nous aurons besoin pour notre système de courrier électronique.

```
apt-get update && apt-get install fetchmail procmail mutt urlview postfix  
msmtp
```

Ensuite, quatre étapes simples suffisent pour configurer les fichiers et nous sommes prêts à utiliser un système de courrier électronique flambant neuf.

Important : À la fin de chaque étape, vous pourrez effectuer un test pour vous assurer que la configuration est correcte.

En d'autres termes, vous aurez un système de courrier électronique totalement fonctionnel une fois ces étapes franchies.

Fetchmail

Fetchmail rapatrie le courrier à partir de serveurs distants et le transmet au système de diffusion de vos machines locales.

Pour l'utiliser, vous aurez besoin de créer un fichier `.fetchmailrc` dans votre répertoire personnel.

Exemple de fichier `.fetchmailrc`

```
touch ~/.fetchmailrc
```

```
poll mail.fai.net protocol pop3 user "mon_nom" password "mon_mot_de_passe"  
## fai : votre fournisseur d'accès Internet.
```

Une fois le fichier `.fetchmailrc` créé, il faudra changer ses droits d'accès avec [la commande chmod](#).



Le fichier ne doit pouvoir être lu que par son propriétaire.

Modifiez les droits d'accès comme suit :

```
chmod 600 .fetchmailrc
```

Pour voir **fetchmail** en action, utilisez l'option `-v` (mode bavard).

Pour rapatrier tous les messages, utilisez l'option `-a`.

L'option `-m` demande à fetchmail de transmettre le courrier rapatrié à **procmail**.



Attention : Lors de vos essais, c'est une bonne idée d'utiliser l'option `-k` qui demande à **fetchmail** de laisser le courrier sur le serveur après l'avoir téléchargé.
En cas de problème, votre courrier ne sera pas perdu.

Faire un alias pour fetchmail

Rajouter cette ligne dans le `~/ .bashrc` ou dans le `~/ .zshrc`

```
alias fetchmail="fetchmail -v -m \"/usr/bin/procmail -d %T\""
```

Commentaire :

- -v est le mode verbose et
- -m envoie les messages à procmail pour effectuer le tri

Aperçu

Screenshot de pop mail avec fetchmail :



Procmail

Procmail a pour but de filtrer le courrier que lui envoie **fetchmail**.

Il agit aussi en tant que MDA en délivrant le courrier dans vos boîtes aux lettres (BAL) où **mutt** (le client de messagerie électronique) ira le lire.

Pour utiliser **procmail**, il vous faudra créer un fichier dans votre répertoire personnel :

```
~/ .procmailrc
```

Commençons un fichier `.procmailrc` très simple.



Les règles de filtrage de procmail sont appelées « recipes » (recettes en anglais).
J'ai inclus quelques recettes anti-spam. 😊

Exemple de .procmailrc

[.procmailrc](#)

```
MAILDIR=$HOME/Mail
LOGFILE=$HOME/.procmaillog
LOGABSTRACT=no
#VERBOSE=on (Est utilisé uniquement pour le débogage.)
VERBOSE=off
DEFAULT=$MAILDIR/inbox
```

```
#####
### Spam à la poubelle. ###
#####
:0
* ^Subject:.*(credit|cadeau|cash|money|debt|sex|sale|loan|spam)
$MAILDIR/Spam
```

```
#####
### Plus de messages HTML. ###
#####
:0
* ^Content-Type:.*html
$MAILDIR/Junk
```

```
#####
### Met les messages dans leurs bonnes boîtes. ###
#####
```

```
#####
### Messages des serveurs ###
#####
```

```
:0
* ^From:.*(Serveur1)
$MAILDIR/Serveur1
```

```
#####
### Messages des services système ###
#####
```

```
:0
* ^From:.*(Cron|Anacron)
$MAILDIR/Cron
```

```
#####
### Contact professionnel ###
#####
```

```
:0
* ^Subject:.*(viadeo)
$MAILDIR/Viadeo
```

```
#####
### Université ###
#####
```

```
:0
* ^From:.*(IUT|iut)
$MAILDIR/IUT
```

```
#####
### Appartement ###
#####

:0
* ^From:.*(Le_Proprio)
$MAILDIR/Le_Proprio

#####
### Boite perso ###
#####

:0
* ^From:.*(Gutts)
$MAILDIR/Gutts

#####
### Informatique ###
#####

:0
* ^Subject:.*(ldlc)
$MAILDIR/LDLC

:0
* ^Subject:.*(PC|info|gentoo|debian|awardspace)
$MAILDIR/Informatique

#####
# Dernière règle : tout ce qui n'est      #
# pas trier va dans la boîte par défaut.  #
#####

:0
* .*
$DEFAULT
```

 Étant donné que **procmail** va créer les fichiers de BAL lorsqu'il en aura besoin (en utilisant les noms donnés dans les lignes d'action), il suffit d'affecter \$HOME/Mail à la variable MAILDIR. Pour plus d'informations : <http://www.procmail.org/>

Vous pouvez maintenant tester votre `.procmailrc` avec la commande `fetchmail` que nous avons vue à la première étape.

 N'oubliez pas l'option `-k` pour laisser les courriers sur le serveur, au cas où vous auriez besoin de refaire des tests.

Procmail test

```
fetchmail -akv -m "/usr/bin/procmail -d %T"
```

Maintenant que **fetchmail** et **procmail** ont été exécutés, allez dans `$HOME/Mail` et lisez vos messages avec `less` ou votre gestionnaire de fichiers.

Configuration

Mutt sert à lire et à composer du courrier électronique. C'est un logiciel puissant et personnalisable tout en restant petit et rapide.

Mutt lit et écrit selon quatre formats de BAL :

- mbox,
- MH,
- MMDF et
- maildir.

Le type de BAL est détecté automatiquement.

Dans notre cas, nous utilisons le format **mbox** selon lequel tous les messages d'une boîte sont stockés dans un même fichier.

Mutt peut aussi utiliser des dossiers situés sur un serveur IMAP distant.

Pour plus d'informations sur cette fonctionnalité, consultez « IMAP support » dans la section 4.11 du manuel d'utilisation de Mutt ainsi que son site web <http://www.mutt.org/>.

Lorsque vous avez installé **mutt** à la première étape, un fichier de configuration système a été installé ici `/etc/mutt/Muttrc`

Vous aurez également besoin de créer un fichier :



```
touch ~/.muttrc
```

dans votre répertoire personnel.

Exemple de fichier .muttrc

```
# ~/.muttrc
# Fichier de configuration de Mutt
```

```
# Plus d'information dans "man muttrc"
```

```
#
# 1. Configuration générale
```

```
#

# Le contenu du champ From est généré à partir du nom complet et de
# l'adresse électronique de l'expéditeur, généralement de la forme :
# From: Prénom Nom <adresse_e-mail>
# L'adresse électronique d'expéditeur (par défaut, le contenu de la
# variable d'environnement EMAIL)
set from="adresse@fai.ext"

# Le nom complet d'expéditeur (par défaut, le champ nom complet de
# /etc/passwd)
set realname="NOM Prénom"

# Pour l'envoi de mail
set sendmail="/usr/bin/msmtp"

# Mutt doit générer un en-tête from. Contrairement à ce que raconte
# le manuel, c'est désactivé par défaut !
set use_from

# Liste de toutes mes adresses e-mail (ce paramètre est notamment pris
# en compte quand on fait un "group reply")
alternates adresse_email@01.fr adresse_email@02.com

# Nom de domaine du système où tourne Mutt.
# Il est utilisé pour :
# - compléter les adresses électroniques incomplètes (sans @domaine);
# - générer les en-têtes Message-Id.
# Par défaut : le contenu de /etc/mailname.
set hostname=localhost

# Pour compléter les adresses incomplètes avec ce domaine?
set use_domain

# Le fichier contenant la signature
set signature=~/.signature

# Le fichier contenant les alias
#set alias_file=~/.aliases
#source ~/.aliases

#
# 2. Configuration des mailboxes
#

# Le répertoire qui contient les mailboxes
set folder=~/.Mail

# Pour avoir la liste des mailboxes, on demande à find(1) de lister
# les mboxes et les maildirs, c'est à dire :
```

```
# - les répertoires parents des répertoires nommés 'cur';
# - les fichiers réguliers non enfants de répertoires nommés 'cur',
#   'new' ou 'tmp' (sinon, ça listerait comme boîtes aux lettres
#   tous les messages contenus dans les maildirs).

mailboxes `find ~/Mail -type d -name 'cur' -printf '%h ' -prune -o -type d
-name 'new' -prune -o -type d -name 'tmp' -prune -o -type f -printf '%p '`

# Ne pas lister les boîtes de messages envoyés et ajournés
#unmailboxes '+sent' "+postponed"

# Mailbox contenant les mails envoyés (chemin relatif)
set record=~/.Mail/sent

# Mailbox contenant les mails ajournés (chemin relatif)
set postponed=~/.Mail/postponed

#
# 3. Options
#

# Editeur de texte pour la rédaction des messages
set editor="vim -c 'set textwidth=72'"

# Jolis titres dans son terminal
set xterm_set_titles=yes
set xterm_title="Mutt with %?m?%m messages&no messages?%?n? [%n NEW]?"

# Ne pas afficher les headers...
ignore *
# sauf un certain nombre que l'on énumère ici
unignore From To Cc Subject Date Reply-To
unignore X-Mailer User-Agent
unignore X-Approved-By List-Id X-List
unignore Disposition-Notification-To Return-Receipt-To

# Ordre d'affichage des headers
hdr_order Date From To Cc Subject

# Utilise l'adresse figurant dans le champ Reply-To: quand on répond
set reply_to
# Locale à utiliser
set locale=fr_FR.UTF-8

# Encodages à utiliser
set charset=utf-8
set send_charset=utf-8

# Format de la date à afficher avant les citations quand on répond
set date_format="%A %d %B %Y à %I:%M:%S%p"
```

```
# Les texte à afficher avant les citations quand on répond
set attribution="Le %d, %n a écrit : "

# Répond sans demander confirmation du destinataire et du sujet
# avant de passer à l'édition du message
set fast_reply

# Inclu le message original dans la réponse sans demander confirmation
set include=yes

# Ne pas aller au message suivant quand on a atteint la fin du
# message précédent et que l'on invoque la fonction "next page"
set pager_stop

# Afficher 6 lignes de la liste des messages en haut de l'écran
# quand on lit un mail
set pager_index_lines=6

# Méthode de tri par défaut = par thread
folder-hook . set sort=threads
# Les thread sont triés par date
folder-hook . set sort_aux=date
# La mailbox "sent" est triée par date
folder-hook "sent" set sort=date

# Affiche le nom d'expéditeur contenu dans ~/.aliases
# si l'adresse e-mail se trouve dedans
# set reverse_alias

# Tenir compte du header "Mail-Followup-To"
set honor_followup_to

# Beep en cas d'erreur
set beep

# Beep quand un nouveau message est arrivé
set beep_new

# Ne pas demander confirmation avant d'effacer un message en quittant
# une mailbox (sinon, mettre "ask-yes")
set delete=yes

# Editer les headers quand on compose le message
set edit_hdrs

# Ne se fie pas au champ "Subject" pour construire les threads
# mais uniquement aux champs "In-Reply-To" et "References"
set strict_threads

# Ne marque pas un nouveau message comme "vieux" quand on quitte la mailbox
# dans laquelle il se trouve
```

```
unset mark_old

# Pas de marker pour les lignes wrappées
unset markers

# Faire-suivre un message en l'attachant au format MIME
set mime_forward

# Format de sujet pour les messages que l'on fait suivre
set forward_format="%a: %s"

# Format de l'index des messages
set index_format="%4C %Z %{%b %d} %-15.15F (%4l) %s"

# Adapte l'adresse mail dans le From en fonction du destinataire
(décommenter)
#send-hook '~t ^.*@via\.ecp\.fr$' 'my_hdr From: Robert Dupont
<robert.dupont@via.ecp.fr>'
#send-hook '~t ^toto@free\.fr$' 'my_hdr From: Robert Dupont
<robert.dupont@free.fr>'

#
# 4. Configuration pour GNU PG
#
# Inutile, Debian a fait le travail pour nous (dans /etc/Mutttrc.d).

## Décode les fichiers attachés de type application/pgp
#set pgp_decode_command="/usr/bin/gpg --status-fd=2 %?p?--passphrase-fd
0? --no-verbose --quiet --batch --output - %f"
#
## Vérifie les signatures pgp/mime
#set pgp_verify_command="/usr/bin/gpg --status-fd=2 --no-verbose --quiet
--batch --output - --verify %s %f"
#
## Décrypte les fichiers attachés pgp/mime
#set pgp_decrypt_command="/usr/bin/gpg --status-fd=2 --passphrase-fd 0 --
no-verbose --quiet --batch --output - %f"
#
## Crée un fichier attaché pgp/mime signé
#set pgp_sign_command="/usr/bin/gpg --no-verbose --batch --quiet --
output - --passphrase-fd 0 --armor --detach-sign --textmode %?a?-u %a? %f"
#
## Crée un fichier attaché application/pgp signé (old-style)
#set pgp_clearsign_command="/usr/bin/gpg --no-verbose --batch --quiet --
-output - --passphrase-fd 0 --armor --textmode --clearsign %?a?-u %a? %f"
#
## Crée un fichier attaché pgp/mime crypté
#set pgp_encrypt_only_command="/usr/lib/mutt/pgpwrap /usr/bin/gpg --
batch --quiet --no-verbose --output - --encrypt --textmode --armor --
always-trust -- -r %r -- %f"
#
```

```

## Crée un fichier attaché pgp/mime crypté et signé
#set pgp_encrypt_sign_command="/usr/lib/mutt/pgpwrap /usr/bin/gpg --
passphrase-fd 0 --batch --quiet --no-verbose --textmode --output - --
encrypt --sign %?a?-u %a? --armor --always-trust -- -r %r -- %f"
#
## Importe une clé dans mon trousseau de clés publiques (pubring)
#set pgp_import_command="/usr/bin/gpg --no-verbose --import -v %f"
#
## Exporte un clé depuis mon trousseau de clés publiques (pubring)
#set pgp_export_command="/usr/bin/gpg --no-verbose --export --armor %r"
#
## Vérifie une clé
#set pgp_verify_key_command="/usr/bin/gpg --verbose --batch --
fingerprint --check-sigs %r"
#
## Lit dans le trousseau de clés publiques (pubring)
#set pgp_list_pubring_command="/usr/bin/gpg --no-verbose --batch --quiet
--with-colons --list-keys %r"
#
## Lit dans le trousseau de clés secrètes
#set pgp_list_secring_command="/usr/bin/gpg --no-verbose --batch --quiet
--with-colons --list secret-keys %r"
#
## fetch keys
## set pgp_getkeys_command="pkspycwrap %r"
#set pgp_getkeys_command=""

# pattern for good signature - may need to be adapted to locale!

# set pgp_good_sign="^gpg: Good signature from"

# OK, here's a version which uses gnupg's message catalog:
# set pgp_good_sign="`gettext -d gnupg -s 'Good signature from "' | tr -d
',,,'`"

# This version uses --status-fd messages
# set pgp_good_sign="^\[GNUPG:\] VALIDSIG"

#
# 5. Binding des touches
#

# Generic
bind generic      \e0H      first-entry
bind generic      \e0F      last-entry
bind generic      <up>      previous-entry
bind generic      <down>    next-entry
bind generic      <pageup>  previous-page
bind generic      <pagedown> next-page

```

```
# Index
bind index      s          save-message
bind index      \t         next-unread
bind index      b          bounce-message

# Pager
bind pager      e          exit
bind pager      s          save-message
bind pager      \t         next-unread
bind pager      b          bounce-message
bind pager      \e0H       top
bind pager      \e0F       bottom
bind pager      n          search-next
bind pager      <up>       previous-line
bind pager      <down>     next-line

# Macros
macro index     G          c?\t
macro pager     G          c?\t
macro generic   <f2>       "!mailq"
macro pager     <f2>       "!mailq"
macro generic   <f3>       "!postqueue -f"
macro pager     <f3>       "!postqueue -f"
macro generic   <f4>       "!fetchmail"
macro pager     <f4>       "!fetchmail"
# Pour les utilisateurs de "isync", en remplacement des deux lignes
# précédentes
#macro generic  <f4>       "!isync -a"
#macro pager    <f4>       "!isync -a"

#
# 6. Les couleurs
#

# Couleur de la barre de sélection
color indicator brightyellow red
# Couleur des messages d'erreur
color error     brightred   default
# Couleur des indicateurs en haut et en bas de la fenêtre
color status    brightwhite blue
# Couleur de l'arbre des threads
color tree      brightmagenta default

# Couleur des headers
color header    white       default "^date:"
color header    brightgreen  default "^from:"
color header    brightwhite  default "^to:"
color header    brightblue   default "^cc:"
color header    brightyellow  default "^subject:"
color header    cyan         default "^resent-"
```

```

color header      cyan          default "^X-"
color header      cyan          default "^User-Agent"

# Couleur du texte cité
color quoted      green         default
# Couleur de la signature
color signature   brightred     default
# Couleur des URLs
color body        magenta       default "(ftp|http|gopher|wais|file)://[^(
)<>,]+)"
# Couleur des adresses e-mail
color body        magenta       default "[-a-z_0-9.]+@[-a-z_0-9.]+)"
# Couleur des smileys
color body        brightyellow  default "[:]+[-^o]?[)]>}D]+"
color body        brightgreen   default "([:;]+[-^o]?[Pp]+)|([:;][-
^o]?[)]>}D]+"
color body        brightred     default "[:;]+[-^o]?[[(<{|\|/\\}+)"

```

Pour information, ceci n'est qu'un exemple de fichier `.muttrc`. Vous pouvez configurer de nombreuses autres options, l'intégration de GPG par exemple. Vous êtes maintenant prêt à tester votre fichier `.muttrc`.

Test de `.muttrc`

```
mutt -y
```

Ceci devrait lancer **Mutt** avec un menu qui affiche les BAL que vous avez créées lors du test 2 avec la commande `fetchmail`.

Tapez le point d'interrogation `?` pour obtenir de l'aide sur la navigation dans les BAL.

Screenshot de `.muttrc`

Visualisation des boîtes aux lettres :



En mode lecture de mail :



En mode envoi de mail :



SMTP

Configurer msmtprc

```
nano -w .msmtprc
```

```
account default
host smtp.votre_fournisseur.net
from votre_nom_d'usager@fournisseur1.net
# Consultez la page man pour plus de détails sur les options
d'authentification.
auth login
user votre_nom_d'usager
password votre_mot_de_passe
# Si votre serveur SMTP supporte le cryptage TLS, décommentez la ligne
suivante.
#tls
```

Régler les permissions

Ensuite, attribuez des permissions d'accès sûres à ce fichier :

```
chmod 600 .msmtprc
```

Finalement, éditez ou ajoutez la ligne suivante dans le fichier .muttrc :

```
nano -w .muttrc
```

```
set sendmail="/usr/bin/msmtp"
```

Lancez **mutt** et écrivez-vous un message pour vous assurez que cela fonctionne.

Lisez la page :

```
man msmtp
```

pour connaître l'ensemble des options disponibles et pour voir un autre exemple de configuration.

Arf, que voilà une proposition bien détaillé gutts. Merci !

Commandes

Testé par **smolski** le 01/01/2013-07:13:11 12 13 14 15

Consulter

Pour consulter ses mails lancez :

```
mutt -y
```

Quitter

Pour quitter mutt tapez la lettre **Q**

Lire

Pour lire un mail, au clavier se positionner dessus et le valider

Quitter

Pour quitter un mail tapez la lettre **Q**

Répondre

Pour répondre à un mail:

1. se positionner dessus (ou rester dans sa lecture) et validez la lettre **R**
2. rédigez son texte avec son éditeur favori.
Par exemple : [vim](#)

Envoyer la réponse

Pour envoyer la réponse, enregistrez votre mail dans votre éditeur puis validez la lettre **Y**



Un petit "r" apparaîtra alors dans la colonne du mail en liste.

Supprimer

Pour supprimer un mail, se positionner dessus et validez la lettre **D**



Un "D" majuscule apparaîtra alors dans la colonne du mail en liste.

Annuler

Pour annuler la suppression ou les suppressions, validez la lettre **U** autant de fois que nécessaire.

Rédiger

Pour rédiger un mail lancez mutt :

```
mutt -x
```

Ou si vous êtes déjà dans mutt, validez la lettre **M**

Adresse

Validez le mail puis mettre l'adresse du destinataire "To : adresse du destinataire"

Sujet

Validez puis mettre le sujet "Subject : le sujet du mail"

Écrire

Rédigez le message à la suite de :

```
(End message with a . on a line by itself)
Message à rédiger...
```

Expédier

Pour expédier le message passez à la ligne et validez un simple point "." :

Exemple

Message genre :

```
bonne année à tous !
.
```

Et puis basta, le message est envoyé ! 😊

Consulter

Il faut entrer dans mutt puis dans les répertoires de la boîte mail ainsi : Pour consulter un mail lancez mutt :

```
mutt
```

Commande

Pour passer en mode commande tapez la lettre **C**
Cette prise de commande vous permet de rédiger en bas de la fenêtre les chemins que vous désirez suivre...

```
Open mailbox ('?' for list): ~/Maildir
```

Liste

Pour consulter la liste des répertoires de mutt tapez la lettre **?**

Ou bien tapez 2 fois sur **Tab**

/cur

Pour consulter l'historique entier validez le répertoire /cur.
Vous aurez alors la liste des mails.

Lecture

Pour lire le mail sélectionné avec les flèches **↑** / **↓** du clavier tapez la barre **Espace**
Pour en sortir tapez la lettre **I**

Suite du tuto

À suivre... 😊

— *gutts* 2009/02/10 00:16

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

MDA ou agent de diffusion de courrier

From:
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:
<http://debian-facile.org/doc:reseau:mutt>

Last update: **12/08/2017 10:49**



