**Licence d'informatique**
**Module de C/Unix**

Projet Unix

# MC — Mail Commands

Philippe MARQUET

Octobre 1998

**Précision** Quelques questions sont proposées dans ce sujet afin de vous aidez à appréhender le projet. Le projet ne consiste pas uniquement à répondre à ces questions...

Les commandes développées ici sont inspirées du système `MH` — Message Handler — disponible sous Unix et développé par RAND Corporation et l'Université de Californie.

## 1 Description

Le but est de fournir des outils de manipulations de messages. Nous désirons disposer de commandes permettant de lire un message, détruire un message, déplacer un message d'une boîte aux lettres à une autre, etc.

À l'inverse des systèmes habituels de gestion des messages sous Unix tel la commande `mail`, `MC` n'est pas un système fermé de gestion des messages que l'on doit quitter pour retourner au shell. Tout comme `MH`, `MC` est un ensemble de commandes distinctes pour chacune des fonctionnalités proposées. Il est ainsi possible d'entremêler des commandes `MC` et d'autres commandes Unix. Un autre avantage est la possibilité d'utiliser les commandes `MC` pour construire une interface graphique de manipulation de messages.

## 2 Architecture des boîtes aux lettres et messages

Un message est représenté par un fichier. Un message appartient à une boîte aux lettres. Une boîte aux lettres est un répertoire. L'ensemble des boîtes aux lettres sont rangées dans le répertoire désigné par la variable shell d'environnement `$MCpath` ; par défaut `$HOME/MailC`.

La configuration de `MC` est mémorisée dans le fichier `$MCpath/mc.config` sous forme de lignes

*`variable :valeur`*

Parmi les ¡¡ variables ¿¿ définies, citons :

**`CurrentFolder`** le nom de la boîte aux lettres courante ; par défaut `inbox`.

**Question 1**
Comprenez l'intérêt de mémoriser la configuration de `MC` dans un fichier plutôt que dans une/des variable(s) shell d'environnement.

Une boîte aux lettres est structurée de la manière suivante :
- Un fichier `folder.config` de configuration de la boîte aux lettres. Ce fichier est au même format que le fichier de configuration de `MC`. Il définit entre autres la variable `CurrentMail` qui contient le nom du message courant dans la boîte.

– Les messages. Le message numéro $i$ est normalement dans le fichier de nom `i`. Il peut être dans le fichier `i.del`, le message est alors marqué *détruit*, il sera supprimé au prochain nettoyage de la boîte aux lettres. Mise à part lors des nettoyages, le traitement d'un message marqué *détruit* ne diffère pas de celui d'un message normal. Un message peut avoir déjà été lu/visualisé on non. La variable `UnreadList` du fichier de configuration de la boîte aux lettres mémorise les numéros des messages non encore lus.

Exemple : la boîte aux lettres courante est nommée `toto`. Elle contient 3 fichiers dont le second est marqué *détruit*, le message courant est le premier. Les messages 3 et 4 n'ont pas encore été lus :

```
% echo $MCpath
/home/marquet/MC
% cd MCpath
% ls -F
inbox/          mc.config       toto/
% cat mc.config
CurrentFolder : toto
% cd toto
% ls -F
1               2.del           3               4               folder.config
% cat folder.config
CurrentMail : 1
UnreadList : 3 4
% cat 1
From: levaire@lifl.fr
To: marquet@lifl.fr
Date: Wed Aug 26 14:17:02 MEST 1998
Subject: Sujet d'exam

Hello,

As-tu une idee pour un examen de C/Unix ?
(Quelque chose de pas trop difficile ...)
%
```

# 3 Syntaxe des arguments des commandes

1. Les arguments des commandes utilisent la syntaxe suivante pour désigner les messages :

   | | | |
   |---|---|---|
   | *messages* | :: | *un-message*\|`all` |
   | *un-message* | :: | *numero*\|`cur`\|`last`\|`unread`\|`next`\|`prev` |

   | | |
   |---|---|
   | `all` | désigne tous les messages, |
   | `cur` | désigne le message courant, |
   | `next` et `prev` | désignent respectivement le message suivant et précédent, |
   | `last` | désigne le dernier message, |
   | `unread` | désigne le premier message non lu. |

2. Une boîte aux lettres est désignée par son nom (nom du répertoire sous `$MCpath`). Par exemple dans la commande

   ```
   % mcpack toto
   ```

## 4 Outils pour `MC`

Les commandes suivantes ne sont pas des commandes fournies à l'utilisateur mais des utilitaires disponibles pour le développement des autres commandes.

**La commande `mcgetvar`.** Elle affiche la valeur d'une variable définie dans un fichier de configuration de `MC`. La syntaxe de cette commande est la suivante :

`mcgetvar` *`cfgfile`* *`varname`*

En reprenant l'exemple précédent, nous avons :

```
% mcgetvar /home/marquet/MC/mc.config CurrentFolder
toto
```

La commande `mcgetvar` retourne normalement 0. Si le fichier de configuration n'existe pas ou n'est pas lisible, la commande retourne 1 ; si la variable précisée n'est pas définie dans le fichier, la commande retourne la valeur 2. Exemple en Bourne Shell :

```
% mcgetvar /home/marquet/MC/mc.config toto
% echo $status
2
```

**La commande `mcsetvar`.** Elle mémorise la valeur d'une variable dans un fichier de configuration de `MC`.

**Question 2**
Proposez une syntaxe de la commande `mcsetvar`. Précisez les différentes erreurs et valeurs de retour possibles de la commande. Donner le code Bourne Shell *commenté* de cette commande.

**La commande `mcprintmailfilename`.** Elle transforme un argument désignant un ou plusieurs messages en noms de fichier. La syntaxe est la suivante :

`mcprintmailfilename` *`folder`* *`messages`*

Par exemple :

```
% mcprintmailfilename toto unread
/home/west/marquet/MC/toto/3
% mcprintmailfilename toto next
/home/west/marquet/MC/toto/2.del
% mcprintmailfilename toto all
/home/west/marquet/MC/toto/1
/home/west/marquet/MC/toto/2.del
/home/west/marquet/MC/toto/3
/home/west/marquet/MC/toto/4
```

**Question 3**
Donnez le code Bourne Shell *commenté* de la commande `mcprintmailfilename`.

## 5 Commandes de `MC`

**La commande `mcfolders`.** Elle liste toutes les boîtes aux lettres selon le format suivant :

```
% folders
inbox : pas de messages.
toto :  4 messages (1 delete/2 unread); cur= 1.

TOTAL = 4 messages dans 2 folders.
```

**La commande `mcinc`.** La commande `mcinc` prend en entrée les messages contenus dans le fichier désigné par la variable `MailDrop` du fichier `$MCpath/mc.config` (par défaut `/var/mail/$USER`). Elle les répartit dans la boîte aux lettres désignée par la variable `InboxFolder` du fichier `$MCpath/mc.config` (par défaut `inbox`) au format utilisé par `MC`. De plus, `mcinc` met à jour la variable `UnreadList` de la boîte au lettres et remet à vide le fichier utilisé en entrée.

Chacun des messages contenus dans le fichier utilisé en entrée de `mcinc` débute par une ligne de la forme :

```
From levaire@lifl.fr Wed Aug 26 14:17:02 MEST 1998
```

Notez bien le caractère espace qui suit le mot `From` (à ne pas confondre avec `From:`). Le message est ensuite constitué d'un entête (suite de lignes non blanches) ; d'une ligne blanche et enfin du corps du message.

Parmi les lignes d'entête, les lignes `From:` et `Subject:` précisent l'expéditeur et le sujet du message. Ces informations seront affichées par la commande `mcinc`.

Exemple :

```
% mcgetvar $MCpath/mc.config InboxFolder
toto
% mcgetvar $MCpath/toto/folder.config UnreadList
3 4
% mcgetvar $MCpath/mc.config MailDrop
% grep 'From ' /var/mail/marquet | wc -l
3
% mcinc
marquet@lifl.fr                 Re: Sujet d'exam
duchmol@nowhere.org             Re: info on GTY
ceurstem@fil.univ-lille1.fr     Comptes etudiants licence

3 nouveaux messages dans toto (5-7)
% mcgetvar $MCpath/toto/folder.config UnreadList
3 4 5 6 7
% ls -l /var/mail/marquet
-rw-------   1 marquet  0 Aug 26 16:45 /var/mail/marquet
```

**La commande `mcshow`.** La commande

`mcshow` [+*folder*] [*messages*]

montre les messages de la boîte aux lettres désignée, par défaut la boîte aux lettres courante.

Si une boîte aux lettres est spécifiée, elle devient la boîte aux lettres courante.

Si le message n'est pas spécifié, deux cas peuvent se présenter :

1. Il existe des messages non lus. `mcshow` montre lors le premier message non lu.
2. Il n'existe pas de messages non lus. `mcshow` montre alors le message suivant.

La commande `mcshow` utilise la commande `$PAGER` pour visualiser le message ; par défaut `more`. Le message affiché devient le message courant.

**La commande `mcrmm`.** La commande

`mcrmm` [*messages*]

supprime les messages indiqués : les marque *détruit*. Par défaut le message courant.

**La commande `mcscan`.** La commande

`mcscan` [*folder*]...

liste les premières lignes des messages des boîtes aux lettres indiquée, par défaut la boîte aux lettres courante.

**La commande `mcfolder`.** La commande

`mcfolder` [`-f`] [*folder*]

identifie la boîte aux lettres désignée comme la boîte courante. Si la boîte n'existe pas, la commande `mcfolder` demande confirmation et crée la boîte aux lettres. L'option `-f` force la création de la boîte aux lettres sans demander confirmation.

La commande vérifie que la boîte est cohérente : fichier de configuration existant, message courant et messages non lus existants, etc.

Sans argument, la commande `mcfolder` affiche le folder courant.

**La commande `mcrefile`.** La commande

`mcrefile` [*messages*] *folder*

déplace les messages de la boîte aux lettres courante (par défaut le message courant) vers la boîte désignée. Les messages sont insérés à la suite du dernier message de la boîte destination. Il garde dans la boîte destination les attributs qu'ils avaient dans la boîte origine (non lu, détruit...). Les messages déplacés sont ensuite marqués *détruits* dans la boîte courante.

**La commande `mcpack`.** La commande

`mcpack` [*folder*]...

supprime les messages marqués *détruits* des boîtes aux lettres spécifiées (par défaut la boîte courante). Les messages restants sont renommés séquentiellement. Le fichier de configuration de la boîte aux lettres est mis à jour en conséquence.