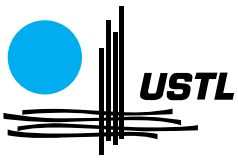

Licence d'informatique
Module de C/Unix

MC — Mail Commands

Examen – Partie Unix

Philippe MARQUET

Septembre 1998

Les documents de cours et TD sont autorisés.
On rendra deux copies séparées pour la partie Unix et pour la partie langage C. La partie Unix est à rendre sur une *copie de couleur.*

Important Plusieurs commandes Bourne Shell sont à développer. On veillera particulièrement à détailler pour chacune des commandes un algorithme avant de donner le code Bourne Shell lui-même. La notation sera répartie entre ces deux composantes de vos réponses.

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
Expliquez 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 :

Id : xam98-sep.tex,v 1.5 2000/11/28 18 :34 :55 marquet Exp

– 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. Donnez 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 `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.

Question 4
Donnez le code Bourne Shell *commenté* de la commande `mcrmm`.

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.

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

Question 5
Donnez le code Bourne Shell *commenté* de la commande `mcfolder`.

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.

Question 6
Donnez le code Bourne Shell *commenté* de la commande `mcpack`.

6 Autres commandes

Seules quelques commandes de base du système `MC` ont été exposées dans le cadre de ce sujet d'examen.

Question 7
Donnez une liste de commandes qu'il vous semble indispensable ou utile d'intégrer à `MC`. Donnez un descriptif rapide et une syntaxe de chacune des commandes proposées.