MH for MM Users

Mary Hegardt Tim Morgan

April 12, 1990

1 Introduction

This document is another in a continuing series on use of the MH mail system at UCI. It is intended for those users accustomed to the MM "user agent" (mail program) under TOPS-20 and who are already familiar with network mail, but who may not be experienced UNIX users. For an introduction to MH, see "MH For Beginners" by Mary Hegardt and Tim Morgan. For complete, detailed information on the MH system, see *The Rand MH Message Handling System: User's Manual* by Marshall T. Rose and John L. Romine. Both documents are available for Xeroxing in suite CS408.

1.1 UNIX Versus Tops-20

The UNIX[1] paradigm is that each command, or program, should perform only one function. An extension of this idea is that the operating system implements only those functions which are necessary, but it does so in a very general way so that programs may still accomplish their functions. This philosophy probably evolved because the original versions of UNIX ran on PDP-11 minicomputers which had only a small memory space for each process.

[1] UNIX is a trademark of AT&T Bell Laboratories

Consequently, all commands in UNIX, with a very few exceptions, are in actuality programs. On TOPS-20, in contrast, many of the most frequently used commands are built into the user's shell, or *exec.* Both the Exec and *csh,* which is typically the user's command interface on UNIX, will accept and parse command lines, attempting to invoke a command as a program if it is not one of the built-in commands. UNIX and TOPS-20 are surprisingly similar internally: the use of the shell, separate processes for each command or program to execute, standard input and output for each program, and many other ideas are common to both operating systems. Users should be familiar with the capabilities of the shell, which is described in the document "Introduction to the Csh."

1.2 The MH User Interface

The MH mail "user agent" is different from most other mail systems. MM, for example, is a *monolithic* system because one program implements all the mail-related functions. The disadvantages of monolithic systems are that (a) they are large, so they tend to put more burden on the computer system, and (b) they allow for much less flexibility. In contrast, MH implements each mail command as a separate program: there is no single program called *mh.* This approach facilitates interspersing mail commands with other, perhaps unrelated, commands.

Another unique feature of MH is that it takes advantage of the facilities provided by the operating system. Most mail agents, such as MM, maintain a file containing the user's mail in a special, usually undocumented, format. When a message is deleted, MM must take care of compacting the mail file. It must be able to distinguish the separate messages contained in the file. MM must also implement a simple text editor to allow the user to enter and modify a message while it is being composed. These functions are essentially those provided by the operating system when separate files are stored within a directory. Therefore, the approach taken by MH is that each message is kept in a separate file. This file simply contains the message, with no other special formatting characters or requirements. All the messages are stored within a normal UNIX directory. This approach makes it easy to add new MH commands, to edit messages using standard text editors, etc.

All your MH related files are stored in a directory within your home directory. Usually this directory is called `Mail` or `mhbox`, although you are free to name it as you choose. Another file in your home directory called `.mh_profile` is equivalent to the `MM.INIT` file under MM. It contains all the options which you prefer for the various MH commands. It also contains the name of the MH directory and the name that you want on your outgoing mail in the `From:` field (your "*signature*").

2 Getting Started

2.1 Incorporating Mail

Another important difference between MM and MH is the concept of the *maildrop* file. Under TOPS-20, the mail transport system delivers new messages directly into the recipient's `MAIL.TXT` file, where they may then be processed with MM. In contrast, the UNIX mail transport system, called *MMDF-II,* makes no assumptions about the user agent used by the recipient. Instead, it puts all new mail into a special file called the *maildrop.* This file is in the `/usr/spool/mail` directory. When you log in, if there is new mail for you in your maildrop, you will be so notified by the message

```
You have new ZOTnet mail -- type inc (or mail)
```

When you are ready to process this new mail, you may type the command

```
% inc
```

("incorporate") which will copy the new mail into separate files, one per message, stored in your "inbox" folder. A folder is a subdirectory beneath your MH directory which is used to store related messages. Additional information on folders is given in Section 4.5, page 13. The "inbox" is a distinguished folder because by default *inc* will always copy new mail into that folder, removing it from the maildrop.

If this is the first time you have used *inc* or any other MH command, the *mh-install* program will inform you that it is creating your `Mail` directory. It will also create the "inbox" folder directory, and `.mh_profile`.

2.2 Message Numbers

As *inc* processes each message, it prints a "scan" listing showing the message number, the date the message was sent, the name of the sender, the subject, and sometimes the initial part of the text of the message. A "scan" listing is thus similar to the output of the HEADERS command in MM. Each message within a folder is given a number, starting with 1, by which it can be referenced. *Inc* will display the numbers assigned to each new message in its "scan" listing.

As in MM, there is a "current message" number which usually identifies the message most recently manipulated by the user. With most MH commands, this will be the default message if no messages are explicitly specified in a command. *Inc* makes the first new message the current message, which is indicated by a "+" character in the scan listing, just after the message number.

Many MH commands take a list of messages to process. A message designation is either a single message number, two message numbers separated by a dash. The dash format indicates a range of messages including the endpoints. A message list consists of one or more message designations separated by spaces. For example, messages 11 through 15 and message 17 may be indicated by typing

```
11-15 17
```

as the argument to some command. There are also several predefined names for messages or lists of messages which may be used in place of message numbers:

cur	The current message (the last one that was handled). Equivalent to "." or "CURRENT" in MM.
next	The next message
prev	The previous message
first	The first message in the current folder.
last	The last message in the folder. Equivalent to % or * in MM.
all	All messages (*first* – *last*). Same as in MM.

It is also possible for you to define your own named *sequences* of messages.

See the *pick* command description for more details.

3 Processing Messages

This section contains a list of the common MM commands and their equivalents in the in MH mail system. A short textual note describes how the MH commands differ from their MM counterparts.

3.1 Listing Messages

As mentioned in Section 2.2, the *scan* command may be used to summarize the messages in a folder, similar to the `HEADERS` command in MM. Unlike most MH commands, however, *scan* defaults to *all* the messages in the current folder unless you specify one or messages on the command line to be scanned. So simply typing

```
% scan
```

is equivalent to typing `HEADERS ALL` (or `H A`) in MM.

3.2 Reading Mail

Unlike the `READ` command in MM, in MH there is no special mail-reading mode (indicated in MM by the `R>` prompt). The command to read messages in MH is *show*. If no message list is specified, then the current message is displayed. The message is displayed by your "showproc", as specified in the `.mh_profile`, described in Section 4.2. Normally, your "showproc" will be *more* or *mhless*. Both of these programs will display your messages one screenful at a time. You press the space bar on your terminal to see the next screenful, or the return key to see the next line.

The command

```
% show next
```

(which will display the first message following the current message in the current folder) can be abbreviated as simply

```
% next
```

Similarly, the command "`show prev`" can be abbreviated as simply "`prev`".

To get a paper copy of a mail message, take the output from the *show* command and pipe it into the *imprint* command.

```
% show 5 | imprint
```

See the manual page for *imprint* for more information.

3.3 Deleting Messages

The equivalent of the `DELETE` command in MM is *rmm* in MH (remove messages). It acts on the current message unless messages are specified on the command line. Unlike MM, the deleted messages will no longer show up in a "scan" listing. But the messages are not completely removed; they are renamed to have a comma prepended to the name of the file containing the message within its folder directory. Therefore, if you need to recover a message, it is possible to go into the directory and rename the message back. Be careful in doing this not to overwrite a new message with the same message number! It is a UNIX convention that files whose names begin with a comma will be removed from disk (*expunged*) early each morning. Therefore, your deleted messages will be available for the rest of the day, unless you remove another message subsequently which has the same message. Then the previously deleted message is gone.

3.4 Replying to Mail

The equivalent of the `REPLY` command in MM is *repl* in MH. *Repl* may be given the number of the message to which you wish to reply, or it will default to the current message. When replying in MM, you are prompted asking if you wish to reply to all the recipients of the message to which you are replying, or only to its sender. In MH, normally the reply will be constructed

to be sent to all the recipients. You may select which recipients receive copies of your reply by using the `-query` option on *repl,* or by putting this option in your `.mh_profile`, as described in Section 4.2. If you wish a reply to go to everyone but yourself, you can use `repl -nocc me`.

3.5 Sending Mail

The equivalent of the `SEND` MM command is *comp* ("compose") in MH. These two commands are fairly similar, except that the recipient of the message cannot be specified on the *comp* command line. The *comp* program invokes a simple editor called *prompter* which will prompt you for the `To:`, `Cc:`, and `Subject:` fields of the message. Then a line of dashes is typed, and you may enter the body of your message (its *text,* in MM terms). When you are finished, type ctrl–D (equivalent to typing ESC or control-Z in MM). Then you'll receive the prompt

```
What now?
```

which is similar to MM's `S>` prompt. You may receive a list of the options that you have at this point by typing "`?`" followed by return. Here is a short list of the options and their meanings. Notice that, unlike MM, there are very few commands to modify the message (such as the `TEXT`, `TO`, `CC`, etc., commands which may be typed at the `S>` prompt in MM). In place of these commands, you use the `edit` command to invoke your favorite text editor on the message, and you use it to make the equivalent changes. You also use your editor to include other files into the body of the message, rather than using control-B, as in MM. One additional use of the `edit` command is for spelling checking. In MM, you may use the command `SPELL` for this purpose. In MH, you type "`edit spell`"[2] instead. This will cause the spelling checker to be run, giving you a list of the possibly misspelled words in your message.

`edit` *editor* — Edit the message using the specified editor. When you exit, you will be back at `What now?`.

[2] Actually, any program named after the "`edit`" command will be invoked with whatever arguments you have given and a path to the file containing the message you are editing.

`list`	Shows the message you just typed
`whom -check`	Verifies that the addresses you have used are valid as far as our system can tell
`send`	Sends the message to the recipients
`push`	Sends the message in the background
`quit`	Quits without sending the message. Saves the text of the message as a "draft". Type `comp -use` to get back to that draft later.
`quit -delete`	Quit, throwing away the draft

3.6 Forwarding Mail

The *forw* command is used in MH to forward messages. It will take a list of messages on the command line to be forwarded, or it will default to the current message if none are specified. It will prompt you like *comp* does for the `To:`, `Cc:`, and `Subject:` fields. Note that, unlike MM's `FORWARD` command, *forw* will not construct a subject line automatically. Also as with *comp,* you will have the opportunity to add additional text to the message(s) which you are forwarding, ended with a control-D.

3.7 Resending Mail

The equivalent of the `RESEND` command in MM is the *dist* ("distribute") command in MH. *Dist* works very much like the *forw* command, except that the prompts will be `Resend-To:`, `Resend-Cc:`, etc. After filling in the headers, a line of dashes is typed giving the impression that additional text can be entered. Nothing could be further from the truth; if you add any text at this point the *dist* will fail. Your only opportunity to add text is in the `Resend-Note:` field.

4 Advanced Topics

4.1 Selecting Messages

In MM, you may use several reserved command words to select messages in place of an explicit list of message numbers. For example, you can type "`DELETE FROM SMITH`" to remove all the messages from a user named "Smith". Rather than building such a capability into each MH program which can process message lists, a special program called *pick* is used instead. Just as there are predefined sequences of messages, such as "`all`", "`cur`", etc., you may use *pick* to define your own sequences. *Pick* is capable of selecting messages from a folder based on the `To:`, `From:`, `Subject:`, `Cc:`, or `Date:` fields, or by searching the body of the message. The patterns to be searched for may include full regular expressions (see the "man" page for *ed(1)* for more information) or simple strings.

Pick may be used in one of two ways. First, it may output the sequence of message numbers which match the search parameters. Using the backquoting mechanism of the shell, these message numbers may then become the arguments to other MH programs. The second way to use *pick* is to have it define a new sequence name which will be the messages which were selected. Only this second method of using *pick* will be described here; see *pick(l)* if you wish to use the first method.

In your `.mh_profile`, add the line

```
pick:  -seq sel
```

Then each time you use the *pick* command, it will define the resulting sequence of messages to be called "sel". Then to "pick" all the messages in the current folder which are from "Smith", just type

```
% pick -from smith
```

To see a summary of those messages, type

```
% scan sel
```

Then to the remove the messages, type the command

```
% rmm sel
```

You can *pick* messages according to any of the headers (`-to -from -subj -cc` or `-date`) or just search all the messages for a given word (`-search`).

4.2 Customizing Your Mail Environment

In MM, you use the `PROFILE` command to tailor your mail environment. This command writes a file called `MM.INIT` in your home directory which is then read by subsequent executions of MM. In the MH system, the file `.mh_profile` serves the same purpose. It is edited with any normal text editor, rather than using a special-purpose command to modify it. The format of the file is line oriented, one line per MH program or MH option to be set. The only required line in the profile is the name of the primary MH mail directory, which is by default `Mail`. This information is specified by the line

```
Path:  Mail
```

The textual name you would like to have on your outgoing mail is specified by the `Signature:` line. For example,

```
Signature:  Mary Hegardt
```

The BBoards which you like to read should also be listed in the `.mh_profile` (see Section 4.6, page 14, for additional information). For example, if you read the "system" BBoard (where all important announcements are posted), as well as "whimsey" and "imagen-users" BBoards, your `.mh_profile` should contain the line

```
bboards:  system whimsey imagen-users
```

Other options may be specified on a per-program basis. The format for these lines is the same. First, the program name is given followed by a colon. Then any flags which are to be the default options for that program are given. Here is a short list of the most common options which you may want to set in your `.mh_profile`:

```
showproc:  mhless
```

The *showproc* is the program used to show messages to you. By default, it is the *more* command. *Mhless* is the same as *more* except that it omits the headers of the messages which you indicate that you wish not to see. Type

```
% man mhless
```

for more information about this program.

```
msh:  -scan
```

Selecting this option causes an automatic scan of new messages on BBoards to be made when reading BBoards with *bbc,* similar to the scan listing produced by *inc.*

```
repl:  -query
```

causes *repl* to ask for each address in the message being replied to if it should be included in the `To:` or `Cc:` fields of the reply being composed.

```
pick:  -seq sel
```

This line will cause messages "picked" by the *pick* command to be put into a sequence named "sel". This sequence name may then be used just as the built-in sequences ("last", "first", etc.).

4.3 Aliases

Using MH, you may specify your own private mail aliases. This feature allows you to store lists of addresses or long internet addresses of people with whom you frequently correspond in one file, and then to address them using short mnemonic names. Typically, you will call your alias file "`aliases`"; it must be stored in your MH directory. The format of this file is simple. The alias is given, followed by a colon, followed by one or more legal mail addresses separated by commas. For example, you might for some reason have an alias for all the users named "Rose" in the ICS department:

```
roses:  prose, srose, mrose, drose
```

In addition to your "`aliases`" file, you will need to modify your `.mh_profile` in order to use aliases. You should add the flag "`-alias aliases`" to the

entries for the commands *ali, whom, send,* and *push,* creating entries for these programs if they aren't already in your `.mh_profile`. Now, messages addressed to "`roses`" will be distributed to all the people listed in the alias.

The *ali* command is used to show you what an alias expands to. You just type

```
% ali alias
```

and *ali* will respond with the expansion of the *alias. Ali* searches the system aliases file in addition to your private ones.

4.4 Blind Lists

There are two different types of so-called "blind addressing" of messages. Users of MM may already be familiar with the "Blind Carbon Copy", or `BCC:` field. It allows you to add recipients to your message just like those who are CC'd, but the normal recipients will not see that the BCC recipients were copied on the message, their replies will not go to the blind recipients, and the blind recipients cannot (easily) reply to the message.

The second type of blind mailing is actually called a "group address list", although it is commonly referred to as a "blind list". The format of this type of address is

phrase: *address_list*;

where the "*phrase*" is any English phrase of one or more words, and the *address_list* consists of one or more addresses separated by commas. The recipients of a message addressed in this fashion will see simply

phrase: ;

so when they reply to the message, their reply will come only to the sender (or the `Reply-To:` field, if one was specified), rather than going to all the recipients of the original list. For example, to use a group address list for the "`roses`" alias you would type:

```
To:  People Named Rose:  roses;
```

This type of group address is very useful for making up lists of related people, such as all the people working on a particular research project.

4.5 Folders

As mentioned previously, folders are directories within your MH directory used to store related messages. There is no equivalent of the folder concept in the MM system. Usually, you will use only the folder "inbox", so you won't have to worry about folders. However, if you process a large volume of mail, then folders become invaluable in managing the messages which you wish to keep for future reference.

Just as there is a "current message," MH maintains a "current folder," which will normally be "inbox". You can change folders either by specifying the folder on the command line of MH programs which take a list of messages as an argument, or by using the `folder` command:

```
% folder +folder_name
```

In general, the folder name is indicated by a "+" sign followed immediately by the folder name, all preceeding any list of messages. For example, you may read the most recent message in a folder called "job_offers" using the command

```
% show +job_offers last
```

This command will have the side-effect of setting the current folder to be "job_offers". You may move messages from the current folder into the "job_offers" folder using the command

```
% refile +job_offers messages
```

where, as usual, the *messages* list will default to the current message in the current folder if none are specified. Note that, in contrast with the *show* command and most other MH commands, the *messages* are *not* considered to be in the folder "job_offers". You may obtain a summary of all your folders by typing the command

```
% folders
```

When you remove messages from a folder using the *rmm* command, the deleted messages will show up as a "hole" in the message numbering. The command

```
% folder -pack
```

will cause all the messages within one folder to be renumbered starting with 1. Similarly, the command

```
% folders -pack
```

will do the same thing for all your folders.

To remove an empty folder, use the command

```
% rmf +folder
```

4.6 Reading BBoards

Two special-purpose programs are utilized in reading BBoards. The first is *bbc,* which is used to check a list of BBoards for new messages. The list of messages may be given on the command line, or if not, it will be taken from the `BBoards:` list in your `.mh_profile`. You may obtain a list of all the available BBoards by typing the command

```
% bbc -topics
```

For each BBoard with unseen messages, *bbc* will invoke the MH shell, *msh,* whose prompt is

```
(msh)
```

The *msh* program allows you to read BBoard mail as if it were normal messages in one of your folders. Almost all the MH commands will work just as the normally do. Typing the command "`quit`" to *msh* causes it to stop reading the current BBoard and go on to the next one containing unseen messages, or to exit if there are no more such BBoards. Typing control-D causes *msh* to exit unconditionally. The command `mark` followed by a message number causes *msh* to act as if you have seen that message and all previous ones.

4.7 Checking for Mail

Under UNIX, there are about a dozen different ways to check for new mail. The easiest way to do it is to set the *csh* variable named "mail" to tell *csh* to check for new mail for you periodically. To do this, add the line

```
set mail=(60 /usr/spool/mail/$USER)
```

to your `.login` file in your home directory. This command says to check for mail if *csh* is about to prompt you with a `%` sign, and if it has been at least 60 seconds since it last checked for mail. The advantage of this method of mail notification, besides simplicity, is that you will never be interrupted by a mail notification. You will only be notified of new mail when you are between commands, when the shell is about to prompt you.

If you desire asynchronous mail notification, which will print to your terminal regardless of what you are currently doing, you may make use of a "Receive Mail Hook" called "rcvtty". To do this, create a file in your home directory called "`.maildelivery`". In this file, put the line

```
* - pipe R /usr/uci/lib/mh/rcvtty
```

Then each time new mail arrives, you will receive a one-line "scan" listing of the mail if your terminal is world-writable. For more information on "maildelivery" files, type:

```
% man 5 maildelivery
```

4.8 Saving Drafts

Normally when you use *comp,* it creates the message being composed in a file called "`draft`" in your MH directory. If you use the "`quit`" option at the "`What now?`" prompt, this file will remain there. You may later use the command

```
% comp -use
```

to resume composing the message.

If you begin composing a new message and there is already a "draft" file, you will be asked for the disposition of this file. Typing `?` return will give you a list of the options at this point. Normally you will either `replace` (delete) the old draft and begin a new one or `use` the old one.

The `-file` switch to *comp* may be used to specify the name of a draft other than "`draft`". For example, one might type

```
% comp -file mary
```

to begin composing a message maintained in the draft file "`mary`". Typing

```
% comp -file mary -use
```

would cause *comp* to resume composing this same draft after a "`quit`" command to the "`What now?`" prompt.

Very advanced users of MH maintain multiple draft files in a *draft folder.* This is a normal folder which holds all your drafts, rather than having just one draft in your MH directory named "`draft`". If you feel that you need to use draft folders, you should consult the *MH User's Manual* for additional information.