NAME

mbox − Format for mail message storage.

DESCRIPTION

This document describes the format traditionally used by Unix hosts to store mail messages locally. **mbox** files typically reside in the system's mail spool, under various names in users' Mail directories, and under the name **mbox** in users' home directories.

An **mbox** is a text file containing an arbitrary number of e-mail messages. Each message consists of a postmark, followed by an e-mail message formatted according to **RFC822**, **RFC2822**. The file format is line-oriented. Lines are separated by line feed characters (ASCII 10).

A postmark line consists of the four characters "From", followed by a space character, followed by the message's envelope sender address, followed by whitespace, and followed by a time stamp. This line is often called From_ line.

The sender address is expected to be **addr-spec** as defined in **RFC2822** 3.4.1. The date is expected to be **date-time** as output by **asctime(3)**. For compatibility reasons with legacy software, two-digit years greater than or equal to 70 should be interpreted as the years 1970+, while two-digit years less than 70 should be interpreted as the years 2000-2069. Software reading files in this format should also be prepared to accept non-numeric timezone information such as "CET DST" for Central European Time, daylight saving time.

Example:

```
 >From example@example.com Fri Jun 23 02:56:55 2000
```

In order to avoid misinterpretation of lines in message bodies which begin with the four characters "From", followed by a space character, the mail delivery agent must quote any occurrence of "From " at the start of a body line.

There are two different quoting schemes, the first (**MBOXO**) only quotes plain "From " lines in the body by prepending a '>' to the line; the second (**MBOXRD**) also quotes already quoted "From " lines by prepending a '>' (i.e. ">From ", ">>From ", ...). The later has the advantage that lines like

```
 >From the command line you can use the '−p' option
```

aren't dequoted wrongly as a **MBOXRD**-MDA would turn the line into

```
 >>From the command line you can use the '−p' option
```

before storing it. Besides **MBOXO** and **MBOXRD** there is also **MBOXCL** which is **MBOXO** with a "Content-Length:"−field with the number of bytes in the message body; some MUAs (like **mutt**(1)) do automatically transform **MBOXO** mailboxes into **MBOXCL** ones when ever they write them back as **MBOXCL** can be read by any **MBOXO**-MUA without any problems.

If the modification-time (usually determined via **stat**(2)) of a nonempty **mbox** file is greater than the access-time the file has new mail. Many MUAs place a Status: header in each message to indicate which messages have already been read.

LOCKING

Since **mbox** files are frequently accessed by multiple programs in parallel, **mbox** files should generally not be accessed without locking.

Three different locking mechanisms (and combinations thereof) are in general use:

- **fcntl**(2) locking is mostly used on recent, POSIX-compliant systems. Use of this locking method is, in particular, advisable if **mbox** files are accessed through the Network File System (NFS), since it seems the only way to reliably invalidate NFS clients' caches.
- **flock**(2) locking is mostly used on BSD-based systems.
- Dotlocking is used on all kinds of systems. In order to lock an **mbox** file named *folder*, an application first creates a temporary file with a unique name in the directory in which the *folder* resides. The application then tries to use the **link**(2) system call to create a hard link named *folder.lock* to the temporary file. The success of the **link**(2) system call should be additionally verified using

stat(2) calls. If the link has succeeded, the mail folder is considered dotlocked. The temporary file can then safely be unlinked.

In order to release the lock, an application just unlinks the *folder.lock* file.

If multiple methods are combined, implementors should make sure to use the non-blocking variants of the **fcntl**(2) and **flock**(2) system calls in order to avoid deadlocks.

If multiple methods are combined, an **mbox** file must not be considered to have been successfully locked before all individual locks were obtained. When one of the individual locking methods fails, an application should release all locks it acquired successfully, and restart the entire locking procedure from the beginning, after a suitable delay.

The locking mechanism used on a particular system is a matter of local policy, and should be consistently used by all applications installed on the system which access **mbox** files. Failure to do so may result in loss of e-mail data, and in corrupted **mbox** files.

FILES

/var/spool/mail/$LOGNAME
$LOGNAME's incoming mail folder.

$HOME/mbox
user's archived mail messages, in his **$HOME** directory.

$HOME/Mail/
A directory in user's **$HOME** directory which is commonly used to hold **mbox** format folders.

SEE ALSO

mutt(1), **fcntl**(2), **flock**(2), **link**(2), **stat**(2), **asctime**(3), **maildir**(5), **mmdf**(5), **RFC822**, **RFC976**, **RFC2822**

AUTHOR

Thomas Roessler <roessler@does-not-exist.org>, Urs Janssen <urs@tin.org>

HISTORY

The **mbox** format occurred in Version 6 AT&T Unix.
A variant of this format was documented in **RFC976**.