NAME

decomposemail − decompose mailboxes into their component email messages

SYNOPSIS

decomposemail [**−hrv**] [**−c** *num*] [**−l** *num*] [**−s** *num*] [*mbox...*]

DESCRIPTION

decomposemail disassembles email mailboxes into their component messages, storing each message in a separate serialized file. These message files can then be reassembled into new mailboxes sorted by date by a program such as **recomposemail**(1), or via standard UNIX commands.

OPTIONS

The following options are supported:

−c Split the mailbox into *num*-message chunks, does not sort.

−h Help, just print an informational message, and exit.

−l Limit, stop processing after *num* messages, does not count messages skipped with **−s**.

−r Recompose, ignored with **−c**. The new mailbox file created will have the suffix ‘‘.sort’’.

−s Skip the first *num* messages.

−v Verbose, provide more feedback.

OPERANDS

The following operands are supported:

mbox The name (with optional path) of a standard Internet style mailbox file. ‘‘−’’ can be used for *stdin* (standard input), which is also the default if no *mbox* argument is given.

USAGE

decomposemail is used to break up standard Internet style mailbox files into serialized files, each holding one email message. This is most useful when used with other programs such as **recomposemail**(1) or **sortmail**(1) to reassemble the message files into new mailboxes. The serialization used provides for easy ordering by date/time. All work is done in the current directory.

With **−r**, a quick and dirty reassembly − sorted by date/time − is performed with no cleanup.

When *mbox* arguments are specified that have suffixes ‘‘.Z’’ or ‘‘.gz’’ indicating that the files are compressed with either **compress**(1) or **gzip**(1), these files will be uncompressed as they are read in.

EXAMPLES

Break out the messages in the mailbox ‘‘zippy’’ into individual serialized files:

decomposemail zippy

Break out the messages in the mailboxes ‘‘../listA’’ and ‘‘../listB’’ into individual serialized files, then recombine them into two new mailboxes ‘‘listA.sort’’ and ‘‘listB.sort’’, each sorted by date/time. There will be no cleanup of the message files. Per message progress will be reported:

decomposemail −rv ../listA ../listB

Break out messages 201 through 300 in the mailbox ‘‘outbox’’ into individual serialized files:

decomposemail −s 200 −l 100 outbox

Break out messages from the standard input, and chunk them into serialized mailboxes each holding 50 messages. The new mailboxes will each have the base name ‘‘stdin’’:

cat mbox.∗ | decomposemail −c 50

FILES

mbox.YYYYMMDD.hhmmss.nnnnn

Serialized files created for each individual message in a mailbox, where ‘‘mbox’’ is the name of the mailbox, ‘‘YYYYMMDD’’ is the year, month, and day, ‘‘hhmmss’’ is the

hour, minute, and second, and ‘‘nnnnn’’ is a serial number. These will be created in the current directory.

mbox-nnnnn Serialized files created for each chunk of *num* messages when the **−c** flag is specified, where ‘‘nnnnn’’ is a serial number starting with ‘‘00001’’. These will be created in the current directory.

DIAGNOSTICS

Returns **0** if no problems encountered.
Returns **non-zero** for any errors encountered.

BUGS

decomposemail is slow.

Data at the beginning of a mailbox that does not appear to be part of an email message will be silently discarded.

decomposemail looks for and uses the GNU version of **awk**(1) where it can find it, but will default to whatever version is available. Since most non-GNU versions have input line length limitations, **decomposemail** will fold input lines that are too long when these other versions are used.

With **−r**, a simplistic algorithm is used to recombine the messages into a new mailbox, and may crash on very large numbers of messages. If this happens, use **recomposemail**(1).

SEE ALSO

compress(1), **gzip**(1), **recomposemail**(1), **sh**(1), **sortmail**(1),
‘‘Toolman: Sorting and Archiving Email’’, *;login:* magazine (USENIX Association), August 1998.

AUTHOR

sortmail was written by Daniel E. Singer (a.k.a. Toolman) at the Department of Computer Science, Duke University.