Identification

Mail facility
K. J. Martin

Purpose

A facility is needed for one user of Multics to leave
messages that can be read later by another user. The
mail command allows a user to append character-string
information to a designated user's mail segment, to read
his own mail segment, and to truncate his own mail segment
to zero length.

Usage

mail key -username- -segname-

where key is one or more of "r", "s" and "d", meaning
that the user wishes to read, send and/or delete mail.

username is the name of the user to receive the mail if
"s" appears in key. The username is of the form
"personal_name.project_name".

segname is the name of a segment to be mailed if "s" appears
in key. The segname argument may be a path name relative
to either the root directory (in which case the first
character is >) or the user's current working directory.
If segname is absent (null character string) and the user
is sending mail, the mail command allows the user to type
in the message directly. In this case, the user may immediately
begin typing in his message. He indicates termination
by a line consisting of only the character specified by
the mode_change option. Default is the period. (Note
the similarity to the context editor - here the user effectively
goes out of an input mode to a final mode. See BX.12
for a discussion of options and BX.9.01 for a description
of the context editor.)

The *-convention described in BX.8.00 may be used for
the username argument. For example a username of *.horror
would designate all users on the horror project.

Discussion

In the mail facility the concept of a home directory for
each user is of some importance. The convention used
to implement this concept is that each user shall have
a directory directly inferior to the user directory directory
(>user_dir_dir) assigned to him. That directory is named
by his user name (personal_name.project_name). If several
users share a directory as "home", that directory has
several names, one corresponding to each user. Thus,
the correspondence of users to directories is not one-to-one;
but for every user, a home directory can be found.

Each user should leave an entry in his home directory
with entry name personal_name.project_name.mail. Otherwise,
a user without append access to that directory will not
be able to append a mail segment. Obviously, a user must
leave the entry around if he wishes to have a special
access list remain in effect for his mail segment. Thus,
to get rid of old mail a user should truncate his mail
segment to zero length using "d" in the key argument to
the mail command.

Implementation

The mail command provides for sending mail to another
user, reading the user's own mail and deleting the user's
own mail.

Before operating on any mail segment, the mail command
interlocks it for writing. After mail is finished, it
unlocks the segment. Interlocking is especially necessary
here because many users may try to access a mail segment
at once. If the user is reading his own mail, (key contains
an "r"), the mail command obtains the name of this user
from the data segment process_info (BY.18), compacts it
to the proper form (no blanks or instance tag), and initiates
the segment >user_dir_dir>personal_name.project_name.>personal_
name.project_name.mail. Mail then checks the bit count
of the segment. If it is non-zero, mail treats the entire
segment as a single character string. It calls the print
command (BX.9.02) to type out the contents on the user's
console. In case of errors, print comments to the user
and returns to the mail command. If the user's mail segment
is empty (bit count of zero or segment non-existent),
mail comments on the error and proceeds to check for other
valid values of key.

The mail command next checks to see if the user wishes
to delete his mail. If a "d" appears in key, the mail ·
command truncates the user's own mail segment to zero
length by calling truncate_seg (BY.2.01). Mail then sets
the bit count on the mail segment to zero.

The mail command last checks to see if key contains an
"s". If so, username must be present; if username is
null, mail considers it an error and comments to the user.
Otherwise, mail then initiates the segment >user_dir_dir
>username>username.mail, creating it if it does not already
exist (providing the sending user has the necessary access).
The mail command assumes that the bit counts of both the
target segment and source segment (if any) are correct.
This assumption is safe in the case of the target segment
since only mail is apt to manipulate these segments.
Any source segment (the seqname argument) prepared using
the context editor (BX.9.01) is also assured to have its
bit count correctly set. Mail first appends a line to
the target mailbox indicating date and who is sending
the mail. Mail calls entryarg (BY.2.04) to process seqname,
then copies the source segment onto the end of the target
segment using write_cs (BY.3.01).

If the user indicates by a null seqname argument that
he wishes to input his message directly to the mail command,
mail calls read_in (BY.4.02) to read from the input stream.
Each line read is appended to the end of the target segment
using write_cs (BY.3.01). The bit count is kept track
of, but is not reset on the branch until the user has
indicated that he is finished. Thus, the target segment
is never left in a half-finished state even if the user
quits. Subsequent accessing of that segment will not
be affected by the extra information which happens to
be on the end.

If the user has used the *-convention on the username
argument, mail calls star (BY.2.08) to obtain the set
of entries to be used. It then appends seqname to each
segment of the set username (or appends the typed-in message
to each segment of the set username).