TO: Distribution

From: P. M. Haber

Date: October 24, 1973

Subject: Proposed ring 1 changes prior to implementation of new "mail" and "send_message" commands.

Please note that with these proposed changes an attempt will be made to define the message segment facility more clearly. Now is the time to make suggestions about what should be included in this facility. Also, various implementation schemes are proposed here. Your comments regarding which you think best and why are requested. Please comment and return on or within three weeks from date of publication.

The Need for New Mail and Send Message Facilities

Both the mail and the send message commands are widely used.. About 70% of the registered users have "mailbox" segments, and over half have "con_msgs" segments. However, there are some rough edges in the current implementation of both commands.

The major objection to the current commands arises from security considerations. The communication segments for both facilities must be readable and writeable by all users allowed to send messages. This means that users other than the intended recipient may read messages, or even "steal" them from the communication segment; that fake messages may be placed in the communication segment, purporting to come from someone other than the actual sender; and that the control information in the communication segment may be accidentally or deliberately destroyed, causing the recipient to loop or find garbage instead of his messages. In addition, because the communication segments are not reliably identifiable by the system, the privileged system functions which wish to communicate with the user do not dare to use either mail or send_message, for fear of encountering a segment which the user is not allowed to write. (For example, if a user placed a link called "mailbox" in his directory which pointed at the password table, and a priviledged system process attempted to send him mail, havoc could result.)

Other minor difficulties exist. The reverse chronological order for messages sent by mail is thought by some to be a mistake. Messages sent by send_message are too often lost during a QUIT by the receiver. In some cases the receiving process loops.

Use of Message Segment Mailboxes

It has been proposed that the message segment become the basis of the new "mail" and "send_message" commands. Using message segments as mailboxes and repositories for sent messages would allow a solution to many of the problems of the current mail facility. The access protection mechanism associated with message segments would allow the owner of a message segment based mailbox to give any user or group of users access to send him mail or messages without simultaneously giving them the right to disturb other messages in the mailbox. A user ring process could not tamper with the sender identification associated with a message segment message. Because the message segment resides in the administrative ring and has associated with it extended access *and a* name suffix unchangeable from the user ring, it would be safely identifiable by system functions. Because messages are threaded within the message segment, a mail command which uses message segments could read messages selectively or in any order desired.

There are, however, several problems associated with using message segments as mailboxes. The major problem has to do with maintaining the integrity of the message segment facility. The mail and send message facility is one "module" which could make use of the message segment facility but would be enhanced by some changes to that facility. It would be useful, for example, to add an additional access bit to the extended access bits signifying "wakeup allowed". This bit would allow the simple implementation of access control for the "allow_message" command. However implementing the addition of this access bit within the message segment primitives would change the .unction of the message segment facility. Similarly, any other module might make requirements of the message segment facility which would cause its definition to change again.

Redefinition of the Message Segment Facility

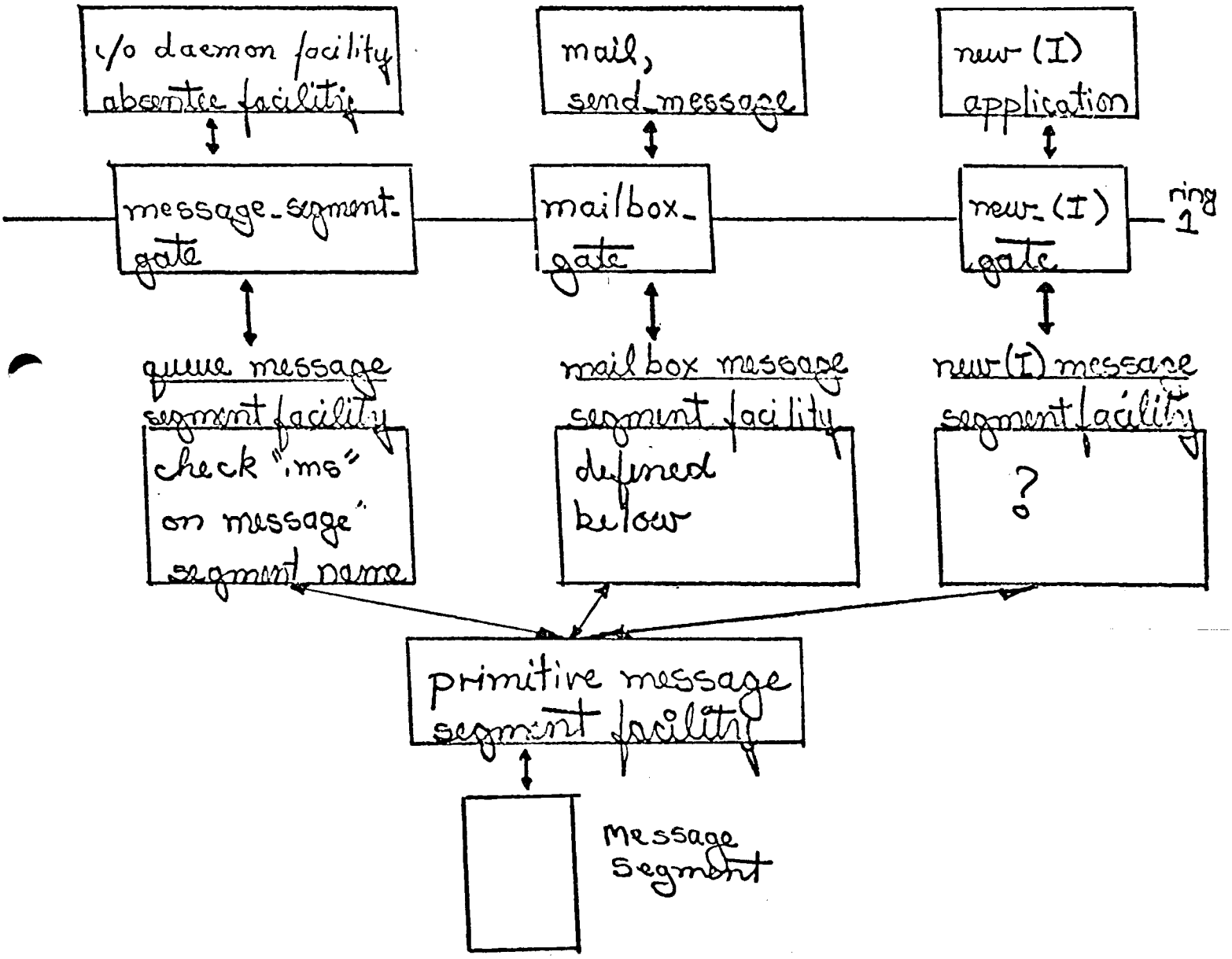
Rather than attempt to modify the message segment primitives to meet the need of any module which might use them, the following scheme is suggested.

1) Move the code specific to the ".ms" name check to an "outer" ring 1 procedure.
2) Reserve bits 9 through 17 of the extended access bits for ring 1 modules which will use message segment primitives.

The message segment facility would be now defined as providing a user with a "primitive message segment" facility plus a ring 1 module which defined a specific application of the primitive message segment. When a new type of message segment was defined, a unique suffix for that type of message segment would be chosen. A ring 1 module would be written which checked to be sure that any operation specified was on a message segment of the correct

type. This same module might also perform additional functions (e.g., access checking, message formatting). The current application of the message segment facility (use of i/o and absentee daemon queues) would define the "queue" message segment facility. This facility would define the queue message segment, a primitive message segment with the suffix ".ms". We would now define the "mailbox" message segment facility.

Pictorially, this proposal may be represented as follows:



Definition of the Mailbox Message Segment Facility

A) Location And Naming Of The Mailbox

Registered users may use the mailbox message segment facility to create a mailbox message segment in their home directory with the name "Person.mbx". Anonymous users may have a mailbox in their project directory with the name "Anonymous.mbx".

B) Additional Extended Access

An additional extended access bit will be employed, designating "wakeup_allowed". If the acl entry of a mailbox corresponding to a user has this bit on and the owner of the mailbox has an entry in the event_table_ (see D below) with the "allow_bit" on, then the user may send the owners' process a wakeup.

C) Information Associated With Each Message

Prefixed to each message will be an priority level and some control information. These variables will be set from the user ring and will be used to implement command options. Their meaning will be defined when the mail and send_message commands are designed.

D) User Wakeup Facility

A facility to allow a message sender to wake up a receivers' process should satisfy the following requirements:

a) A prospective receiving process should be able to allow and defer messages from any user or group of users.

b) The information required to send a wakeup should not be readable in the user ring, so that it will not be possible to get this information and send spurious wakeups to another process.

To satisfy these two requirements the following ring 1 facility is suggested (an alternate proposal is presented in the addendum on page 6):

i) event_table_ -- A table residing in ring one, managed by an event_table_ manager. (This facility must reside in ring 1 to prevent readability of the event channel id in the user ring). Each entry consists of the person-name of the user to whom wakeups may be sent, his project, his process id, his event channel id, a priority level (wakeups will be signalled only if the priority level of a sender's

message is greater than this value), and an "allow bit" (on if the user wishes to allow messages).

ii) event_table_manager_ -- A set of functions for managing the event table.

$add_event - Creates an event channel; creates an event_table_ entry with the allow bit on.

$delete_event - Removes an event_table_ entry; destroys the corresponding event channel.

$send_wakeup - Sends a wakeup to a specified process if allowed.

$set_data - Sets the priority_level and/or the allow bit of an entry in the event_table_.

The above changes will provide the groundwork for commands which will allow the user to say things like "allow the daemons to send me messages; allow any user sending a message of priority >n to send me messages; allow the following people to send me mail; read mail from the following projects...."

After the above changes are implemented, the following course suggests itself.

1) Modify the mail and send_message commands to use message segment mailboxs.

2) Write a user ring subroutine which functions like the mail command but incorporates the new possible options and returns a code.

3) Write new mail command, send message command.

4) Begin having system functions use the facility. For example, when the io daemon reads in a deck of cards on behalf of a user, it could send that user mail or a message telling hime where his input is rather than place a link in his directory pointing to that input as is currently done.

Addendum To Proposal

The following represents a more complex set of changes to the ring 1 message segment primitives and an additional user name/mailbox pathname correspondence table and maintaining facility. These changes would require more time to implement and check out, but would have advantages which will be mentioned as they are described.

1) Alternate Location Of Mailbox

Rather than have a mailbox created in a specific directory, provide a ring 1 facility which creates a mailbox wherever the user wishes and creates an entry in a ring 1 table which maps his name(s) (and initials, if desired) with the pathname of the mailbox created. Provide a delete entry which deletes the mailbox and the mentioned table entry. Modify the current create and delete primitives so they will return an error code if called to operate on a mailbox message segment.

a) advantages

i) users would need to maintain only one mailbox, although they might be registered on more than one project

ii) mail and messages could be sent by person name only, rather than by person-project

Another proposal is to place all the mailboxes in a single directory. It has also been suggested that users not have the ability to delete these mailboxes, thus ensuring system functions that a mailbox for a given user will exist.

a) advantages

i) same as above

ii) existence of a mailbox would not conflict with quota needs

iii) simpler implementation than previous alternative

iv) system functions whould be ensured of .inding a mailbox associated with a user

2) Alternate Wakeup Facility

Rather than place the wakeup information in an event_table_, put it in the header of the message segment mailbox. Provide an entry to the message segment primitives which puts the message in the mailbox and sends a wakeup.

a) advantages

i) Would provide a general feature which absentee and the i/o daemon could also use.

ii) Would eliminate a system data base (event_table_).

b) disadvantages

i) Requires changing the header length of the message segment. Changing the header of a message segment for a specific application is contrary to the notion of a stable message segment facility. However, it might be wise at this point to allocate a few words in the header of the primitive message segment for use by ring 1 message segment applications in general, and make mailbox message segments the first case of such an application.

3) Second Alternate Wakeup Facility

Rather than associate only one event channel with a user in an event table, provide a facility for associating any number of channels with the user. Identify each channel with a character string. For example, if a user wishes to send a wakeup to another user to read his mailbox, he issues a call to send_wakeup which includes the character string "mailbox".

i) advantages

This solution is the most general, providing the handle for any facility to define a ring 1 event channel entry.

(END)