Problem Description

When the mail program is invoked with a control argument such as `-ack` or `-bf`, and no mail path is provided, the program gets a null pointer exception:

```
mail -ack
Error: Attempt by mail$|5217 (>system_library_standard>bound_msg_facility_) to reference through null pointer
r 18:36 0.149 2 level 2
```

This occurs due to the code’s not checking the error code returned from the function `cu_$arg_ptr`.

```
GET_PATH: call cu_$arg_ptr (argno, ap, al, code);
    argno = argno+1;
    if substr (arg, 1, 1) = "-" then go to GET_PATH;
    buffer = arg;
    if buffer = ">" then do;
        code = error_table_$root;
        go to ERROR2;
    end;
    text_length = 0;
    mail_format_ptr = null;
    call user_info_ (sender_name);
```

In the case where mail is invoked with a single control argument (e.g. `-ack`) and no other arguments, the first time the program arrives at the label GET_PATH, it retrieves a pointer to the control argument (“-ack”) in ap, and the length of that control argument in al. It increments argno and then checks to see if the argument (via arg) begins with a “-“, signaling a control argument. If it is (such as the case in point here), it jumps to GET_PATH, calling `cu_$arg_ptr` again, this time with a non-existent argument. The procedure returns `error_table_$noarg` as the value of code, which is ignored. `cu_$arg_ptr` has set ap to null, as there is no argument at this position. The next reference through to arg (through ap) results in an attempt to reference through the null pointer.

The fix is trivial — simply check for a non-zero value of code after the call to `cu_$arg_ptr` and skip the code that attempts to store the path in the variable buffer as well as the code that attempts to see if the path is “>”.

Proposed Changes

Change the relevant code in mail.pl1 to the following:

```c
GET_PATH: call cu_Sarg_ptr (argno, ap, al, code);
    if code ^= 0 then go to SKIP_PATH;
    argno = argno+1;
    if substr (arg, 1, 1) = "-" then go to GET_PATH;
    buffer = arg;
    if buffer = ">" then do;
        code = error_table_$root;
        go to ERROR2;
    end;
SKIP_PATH:
    text_length = 0;
    mail_format_ptr = null;
    call user_info_ (sender_name);
```

The compare_ascii output is shown below:

```c
cpa (lpn mail.pl1) ==

Inserted in B:
B544                if code ^= 0 then go to SKIP_PATH;
Preceding:
A544                argno = argno+1;

Inserted in B:
B552      SKIP_PATH:
Preceding:
A551                text_length = 0;
```

Comparison finished: 2 differences, 2 lines.
r 18:45 0.860 17

Documentation

No documentation is required.

Testing

Testing will involve invoking the mail command with the various combinations of control arguments and non-control arguments.

Version History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-01-21</td>
<td>1.0</td>
<td>Eric Swenson</td>
<td>Initial revision of the MCR.</td>
</tr>
</tbody>
</table>