Problem Description

When Multics crashes early in the boot process, the operator is offered to create an “early dump tape”. These dump tapes can be read into >dumps using the read_early_dump_tape command. The azm program can, in general, read these early dumps, but in some cases, fails. The obsolete command, ol_dump, does not have issues reading these early dumps, but azm faults with the following:

```
Error: out_of_bounds at amu_fdump_translate_$to_temp_seg|610
>system_library_tools>bound_amu_
re referencing >dumps>040519.1142.1.1234|776000
Attempt to access beyond end of segment.
 r 15:31 0.504 525 level 2
```

azm uses the utility procedure amu_fdump_translate_ to read the segments dumped in the early dump tape, and allow referencing of these segments by azm using the segment numbers that these segments had prior to the crash. Specifically, the locate_segment procedure scans the regions in the dump image and attempts to match them to a segment mapping table. However, it does the wrong thing if a segment doesn’t appear in the dump. This is signaled by the variable total_lth’s being set to 0.

It includes the following code:

```
if (offset + total_lth) <= fdump_info.dump_seg_lth (dump_seg_idx) then do;
  lth = total_lth - 1;                 /* image is whole in single dump segment */
  base_2 = null ();
  lth_2 = 0;
end;
```

The above code uses total_lth (which is 0) and sets lth to total_lth - 1, which is -1.

The variable base has been set thus, just prior to the above code:

```
base = pointer (fdump_info.dump_seg_ptr (dump_seg_idx), offset);
```

The locate_segment procedure returns to its caller having set base to a location in the image, and lth to -1.

The caller of locate_segment attempts to copy the segment, using the length of -1 and gets the above-mentioned out_of_bounds error.

The fix is trivial — simply check to see if total_lth is 0, and if so, set base to null(). Callers of locate_segment already properly deal with a base of null, so the above fix lets azm continue without problems.

Proposed Changes

The fix is simple and is illustrated by a comparison of the old and new version of amu_fdump_translate_.pl1:

```plaintext
cpa [lpn amu_fdump_translate_.pl1] ==

Inserted in B:
B252            if total_lth = 0 then do;
B253                    base = null ();
B254                    base_2 = null ();
B255                    return;
B256                 end;
B257

Preceding:
A252            if (offset + total_lth) <= fdump_info.dump_seg_lth (dump_seg_idx) then do;

Comparison finished: 1 difference, 6 lines.
r 17:12 0.752 15

The code sets `base` and `base_2` to `null()` in the case that `total_lth` is zero.

Documentation

No documentation is required.

Testing

I’ve tested the modified azm on several early dumps as well as normal dumps.

Version History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-05-25</td>
<td>1.0</td>
<td>Eric Swenson</td>
<td>Initial revision of the MCR.</td>
</tr>
</tbody>
</table>