The iox_$get_chars subroutine description outlines how a stream I/O module’s get_chars entrypoint should operate. The defect is in the tape_mult_ I/O module. For an iox_$get_chars call where the switch is attached to tape_mult_, n_read is not set to 0 when the next byte position is set beyond the end of file. It contains whatever value the argument passed to the n_read parameter held before the call to iox_$get_chars.

**Entry: iox_$get_chars**

This entry point reads 9-bit bytes from the unstructured file or device to which an I/O switch is attached. The switch must be open for stream_input or stream_input_output. The desired number of bytes, N, is specified in the call. Some I/O modules may actually read fewer than N bytes into the buffer, even though N bytes are available from the file or device. In this case the code error table_$short_record is returned. When this code is returned, the caller may again call the iox_$get_chars entry point to get more bytes. The contents of the buffer beyond the last byte read are undefined.

If the switch is attached to a file, bytes are read beginning with the next byte, and the next byte position designator is advanced by the number of bytes read. If fewer than N bytes remain in the file, the code error table_$short_record is returned, and the next byte position is set to end of file. If the next byte position is already at end of file, the code error_table_$end_of_info is returned.

It is possible to write a program which takes certain actions if a call to iox_ takes longer than a certain amount of time. See the timed_io_$get_chars entry point.

**USAGE**

```plaintext
declare iox_$get_chars entry (ptr, ptr, fixed bin(21), fixed bin(21), fixed bin (35)) ;
call iox_Sget_chars (iocb_ptr, buff_ptr, n, n_read, code);
```

**ARGUMENTS**

- iocb_ptr
  - points to the switch’s control block. (Input)
- buff_ptr
  - points to the byte-aligned buffer into which bytes are to be read. (Input)
- n
  - is the number of bytes to be read where n>=0. (Input)
- n_read
  - is the number of bytes actually read. (Output) If code is 0, n_read equals n.
- code
  - is an I/O system status code. (Output)
The highlighted text in this description pertains to one of the problems reported in Multics Change Ticket: #195.

In that ticket, an exec_com uses io_call to demonstrate that the tape_mult_ I/O module does not handle the iox_$get_chars call correctly when returning with code = error_table_$end_of_file. If a final successful read returns a partial record (n_read < n) and code = error_table_$end_of_file, any subsequent read returns error_table_$end_of_file, but does not set its n_read parameter.

While the correct error code is being returned for reading when positioned beyond the end-of-tape mark, a potentially non-zero value in n_read makes it appear that some data was also returned. In such cases, the buffer pointed to by the buff_ptr parameter is unchanged.

**Proposed Change**

When tape_mult_read_$get_chars is first entered, place zero in the n_read parameter.

The green highlight shows the line being added:

```
pr tape_mult_read_.pl1 -from 183 -for 12 -nb
183  get_chars: entry (Piocb,
184         Pcallers_buffer,
185         number_of_chars_requested,
186         Lcallers_buffer,
187         code);
188
189  /* ********************************************************************** */
190         Lcallers_buffer = 0;
191         code = 0;
192         Sentry_was_at_get_chars = "1"b;
193         call get_number_of_chars_requested (Sentry_was_at_get_chars, code);
```

**Description:**

tape_mult_read_$get_chars fails to set Lcallers_buffer (number of chars read) when called while positioned beyond EOT (end-of-tape) marker. [Bug:195]

**Installation_directory:**  >udd>m>gd>w>tm;

**Build_script:**         tm.mb;

**Bound_obj:**            bound_tape_mult_      IN: hard  UPDATE;
    source:           tape_mult_read_.pl1    REPLACE    compiler: pl1 -ot;

**mbuild:**  compare

```
-------- tape_mult_read_.pl1

compare_ascii
>ldd>hard>source>bound_tape_mult_.s.archive::tape_mult_read_.pl1 ==

... [Changed date format in earlier history comments omitted.]
```
A69  16) change(87-07-07,GWMay), approve(87-07-17,MCR7747),
    audit(87-07-07,Beattie), install(87-07-17,MR12.1-1043):
    Changed to return immediately when records are out of sequence.
B69  16) change(1987-07-07,GWMay), approve(1987-07-17,MCR7747),
    Changed to return immediately when records are out of sequence.
B70   17) change(2020-01-22,GDixon), approve(2020-01-22,MCR10071):
    tape_mult_read_$get_chars fails to set Lcallers_buffer when called
    while positioned beyond EOT (end-of-tape).  [Bug:195]

Inserted in B:
B191            Lcallers_buffer = 0;

B191            Lcallers_buffer = 0;

Preceding:
A188            code = 0;

Comparison finished: 9 differences, 64 lines.

mbuild:

Testing

This code change was tested by creating a short tape containing 10000 characters of data.
The call command is used to read this data from the tape records into a 2900 character
buffer. This ensures that the final read will contain fewer than a full buffer of data.
The first three reads each return with parameters: buff_ptr shows the actual characters read from the tape; n_read = 2900 gives number of characters read into buff_ptr; and code = 0.

io_call attach S tape_mult_GCD001
Mounting tape GCD001 for reading
Mounted Multics volume "GCD001" (recorded at 800 BPI), on device tapa_05
io_call open S stream_input
call iox$_get_chars -in 247|24414 -inout "" -id buff_ptr -addr "char(n)" -in 2900 -id n -inout 9999 -id n_read -out -id code -code

-- Return from: iox$_get_chars -------

buff_ptr 417|2140 ->
....|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8....|....9....|3000
....|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8....|....9....|5800

n_read 2900
code OK

call iox$_get_chars -in 247|24414 -inout "" -id buff_ptr -addr "char(n)" -in 2900 -id n -inout 9999 -id n_read -out -id code -code

-- Return from: iox$_get_chars -------

buff_ptr 417|2140 ->
....|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8....|....9....|5900
....|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8....|....9....|8700

n_read 2900
code OK

call iox$_get_chars -in 247|24414 -inout "" -id buff_ptr -addr "char(n)" -in 2900 -id n -inout 9999 -id n_read -out -id code -code

-- Return from: iox$_get_chars -------

buff_ptr 417|2140 ->
....|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8....|....9....|8800

A fourth read returns with: n_read = 1300, and code = error_table$_end_of_info.

call iox$_get_chars -in 247|24414 -inout "" -id buff_ptr -addr "char(n)" -in 2900 -id n -inout 9999 -id n_read -out -id code -code

-- Return from: iox$_get_chars -------

buff_ptr 417|2140 ->
....|....1....|....2....|....3....|....4....|....5....|....6....|....7....|....8....|....9....|8800
For programs that read data until no further characters are returned, a fifth read should return with: \text{n\_read} = 0, and \text{code} = \text{error\_table\_$end\_of\_info}.

- The installed tape\_mult\_ never changes the \text{n\_read} parameter from its value prior to the call (9999); but it does set \text{code} properly.

\begin{verbatim}
call iox\_$get\_chars -in 247|24414 -inout "" -id buff_ptr -addr "char(n)" -in 2000 -id n -inout 9999 -id n\_read -out -id code -code
-- Return from: iox\_$get\_chars --------
buff_ptr 417|2140 ->
n\_read 9999 
code   \text{error\_table\_$end\_of\_info}  End of information reached.
\end{verbatim}

- The repaired tape\_mult\_ sets \text{n\_read} = 0, and also sets \text{code} properly.

\begin{verbatim}
call iox\_$get\_chars -in 247|24414 -inout "" -id buff_ptr -addr "char(n)" -in 2000 -id n -inout 9999 -id n\_read -out -id code -code
-- Return from: iox\_$get\_chars --------
buff_ptr 417|2140 ->
n\_read 0 
code   \text{error\_table\_$end\_of\_info}  End of information reached.
\end{verbatim}

Notice that both the installed tape\_mult\_ and the repaired program return after the fourth read with the parameter: \text{code} = \text{error\_table\_$end\_of\_info}. The documentation for \text{iox\_$get\_chars} says this should be: \text{code} = \text{error\_table\_$short\_record}.

This variance from the \text{iox\_$get\_chars} documentation is not being changed as part of this repair proposal. Many programs (both installed and privately owned) use tape\_mult\_ to read tapes. Changing the code returned with a short record could adversely affect their operation. At this point, it is prudent to ignore non-use of the \text{error\_table\_$short\_record} return code by tape\_mult\_.

**Documentation**

The code being changed follows the existing documentation, so no documentation changes are anticipated.
## Version History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Author</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-01-22</td>
<td>1.0</td>
<td>Gary Dixon</td>
<td>Initial version of MCR.</td>
</tr>
<tr>
<td>2020-01-23</td>
<td>1.1</td>
<td>Gary Dixon</td>
<td>Correct problems found by Eric Swenson during review of the initial MCR.</td>
</tr>
</tbody>
</table>