3/12/70

new ecs facility

return authorization + block data transfer during call

will permit the return of at most 1 block of data + 15 source objects

1) format original call

The 2 words immediately preceding the list control the return authorization.

A) 1st word preceding is list controls data return

```
  18  18
  C   A
```

up to C words can be returned starting at address A.
[If C=0 no words can be returned; hence a demanded refuses authorization]

B) 2nd word preceding is list controls object return

```
  18  18
  C   A
```

up to C objects can be returned starting at list index A
[as above, C=0 for no authorization]
2) new versions of return & return

IP1 & IP2 will control data and object return
we will also retain old versions of return & return?

A) IP1 controls data return

<table>
<thead>
<tr>
<th>12</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

upto C words will be returned starting at address A

B) IP2 controls object return

<table>
<thead>
<tr>
<th>18</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

upto C objects will be returned starting at CL register A
3) Implementation of new forms of return

If both counts are zero, no action.

If either count is zero, check for errors [c-list flush and field
flush errors]

If no error, copy data and c-list entries to a buffer, exit
perform swap back to calling subprogram.

Clean its return control data, reducing counts to the minimum
it called out caller, now check for c-list flush + field flush
errors, if using, copy data and c-list entries to proper
location.

Now look for interrupt signals!

4) Variation on original call

change format XJT to:

\[ \begin{array}{c|c|c}
  \text{XJT+303} & \text{code} & \text{addr} \\
  \hline
  \text{12} & \text{A} & \text{5} \\
\end{array} \]

S will be skip field if no return [most code now uses 0 or 1]
A address at a 2 word return control in form:

\[
\begin{array}{c|c|c|c|c}
  & 18 & 18 \\
  \hline
  A & C & A_1 \\
  & C_2 & A_2 \\
\end{array}
\]

A, C, address + count of data return authorization
A_1, C_2 address + count of object return authorization
If \( A = 0 \), no authorization, if \( (A+1) = 0 \) no data authorization
\( (A+1) = 0 \) no object authorization.
addition

The es buffer can also be used to implement block data transfer during a call.

need

1) new kind of parameter specification:

2) The IPL entry will be 9 words as follows:

<table>
<thead>
<tr>
<th>18</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>

If C > n, error
else transfer C words from A to the parameter area in called subprocess (via es buffer, of course)

3) The next data word in parameter area of called subprocess will, of course, be n words beyond beginning of this block.