Identification

Print segment linkage information
print_link_info
J. M. Grochow

Purpose

It is often desirable to see a listing of the segment entry information and of the links required by a particular segment. Although the EPLBSA assembler currently produces such a listing, the required information is not readily available for bound segments (except in highly unreadable numerical format). It is also reasonable to assume that a user will not have assembler listings for all segments of interest (e.g., the segment library) and that new compilers or assemblers may not automatically print this linkage information. The print_link_info command may be used in any of these cases.

Usage

The user issues the command:

   print_link_info pathname callname

where both arguments are character strings as follows:

   pathname=pathname of segment (working directory is assumed if no directory name provided)

   callname=name segment is to be initiated under (see below)

print_link_info will either print the entry point and linkage information or print an error message (if it cannot find text and link for the segment). If the segment is initiated (and none of the links have yet been snapped) and if a pointer to a seg_util-like status array (BY.2.12) exists then the user may issue a call from a procedure as follows:

   call print_link_info$ptr (status_ptr);

where status_ptr is the pointer to the status array.

If text and link cannot be found for the segment or for some reason cannot be initiated, an error condition is raised and a message printed to the user.
Implementation

print_link_info will initiate the segment to obtain a fresh copy (except as noted above in a call to print_link_info). If callname is null print_link_info will supply segment management with a unique name under which to initiate the segment (the user should usually not supply callname). If callname is provided, this name will be passed to segment management.

print_link_info calls write_out for each line of output as it is interpreted. Future versions will allow either console print out or file creation for offline printing.

Output

The text segment length is first printed and then each linkage block is interpreted for the following:

1. Entry and segdef names and values
2. Symbolic segment and symbol names for each link pair (and trap work information, if any).

Each entry under "Entry points and segdef names" contains the following:

1. the ASCII representation
2. the value (in octal)
3. the class of the symbol (entry point, external label, text or symbol segment definition).

Each entry under "Link pairs" contains the following:

1. The address of the link pair relative to the base of the linkage segment.
2. The ASCII representation of the segment and symbol to which the link points or an indication that the link points to one of its own segments ("text", "link", or "symbol" is printed) and the numerical offset (in octal).
3. The call pointer and argument pointer (in octal) of the trap word if it exists.
Information on the structure of the various parts of the linkage section can be found in BD.7.01 and BD.7.05. A sample output listing is as follows:

**Text segment length (in octal):** 5426

**Linkage block number 1**

**Entry points and segdef names**

<table>
<thead>
<tr>
<th>Function</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>unbind</td>
<td>42</td>
</tr>
<tr>
<td>getfirst</td>
<td>34</td>
</tr>
<tr>
<td>bind</td>
<td>26</td>
</tr>
<tr>
<td>bug</td>
<td>20</td>
</tr>
<tr>
<td>rel_text</td>
<td>0</td>
</tr>
<tr>
<td>rel_link</td>
<td>357</td>
</tr>
<tr>
<td>rel_symbol</td>
<td>404</td>
</tr>
</tbody>
</table>

**Link pairs**

10  `<datmk_>`[[datmk_]]
12  `*text|1066`
14  `<fsstat>![cm_ptr]`  trap: call 10  arg 12
16  `*text|22`
24  `*text|27`
33  `*text|533`
40  `*text|570`

**Linkage block number 2**

**Entry points and segdef names**

<table>
<thead>
<tr>
<th>Function</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupstat</td>
<td>116</td>
</tr>
<tr>
<td>unassign</td>
<td>110</td>
</tr>
<tr>
<td>entry</td>
<td>102</td>
</tr>
<tr>
<td>rel_text</td>
<td>0</td>
</tr>
<tr>
<td>rel_link</td>
<td>357</td>
</tr>
<tr>
<td>rel_symbol</td>
<td>404</td>
</tr>
</tbody>
</table>

**Link pairs**

56  `<ptr>[[rel]]`
60  `<datmk_>`[[datmk_]]
62  `*text|3016`
64  `<fsstat>[[cm_ptr]`  trap: call 60  arg 62
66  `<ilock>[[looplock]`
70  `<panic>[[panic]`
72  `<bug>[[unbind]`
74  `<bug>[[bind]`
76  `<ilock>[[loopunlock]`
100 `*text|1243`
106 `*text|1250`
114 `*text|2237`

**End of Linkage Information**