Published: 10/13/66
(Supersedes: BE.8.01, 1/14/66;
BE.8.01, 6/28/66)

Identification

Linker for the Pseudo-supervisor
D. H. Johnson, D. Boyd

Purpose

The procedures described in this section establish intersegment references during the execution of a process. External references go indirectly through the linkage section (BD.7.01) of a segment. The first time an external reference is made an fi modifier in the linkage section is recognized by the GE645 hardware during address modification. This causes a fault which executes a pair of instructions in the fault vector locations for fi faults. A fault handler is called which in turn calls the linker procedure. If the referenced segment is not in the segment name table a search procedure is invoked which will dynamically load the segment if it exists in the segment library. The linker changes the fault to an ITS or ITB pair which points to the desired reference. The fault handler, the linker, the search procedure, and another procedure, link_, which calls the linker directly, are described below.

Fault Tag 2 catcher

The two locations in the fault vector associated with the fi or fault tag 2 contain the following instructions:

```
SCU  = its(segment number of f2catc,0),*
TSS  = its(segment number of f2catc,8),*
```

The fault tag 2 catcher, f2catc, is called directly by the TSS instruction in the fault vector. The SCU information at the time of the fault was stored by the previous instruction into the first 6 locations of f2catc. The fault tag 2 catcher stores the scu information, registers, and bases into the stack. It then calls the linker. Upon return from the linker, f2catc determines whether there was an error in linking. If so, it escapes to the 645 simulator after writing error comments into the user's error file and the process is terminated. If the link was set correctly, f2catc will restore the registers, bases, and modified control unit information. Execution then proceeds as before the fault occurred.

If the linker recognizes an error while attempting to set the link, it returns to f2catc with an error code. f2catc will interpret the error code and write comments

into the user's error file which should pinpoint the linkage error. Following is the list of possible linkage errors:

1. Fault occurred in a linkage section with no link definitions.
2. Illegal external reference type code.
3. External symbol definition not found in linkage section.
4. Segment name not found in segment name table or in the segment library and trace procedure (see section BE.12.01) is not on.
5. Linkage section segment of external symbol not found in segment name table.
6. Tried to trap before link with call ptr equal 0.
7. Undefined third argument used in call to linker.
8. Illegal control word returned to linker from trace procedure. (see section BE.12.01)
9. Illegal control word returned to linker from terror procedure. (see section BE.12.01)
10. Fault tag 2 encountered in non-standard situation.

Linker

The procedure to set links is called as follows:

```
pseudo_supervisor$linker( mcdata,ubases, option, error, code)
```

where the arguments are:

	identifier	attribute	meaning
1.	mcdata	pointer	machine conditions at time of fault
2.	ubases	pointer	user bases registers at time of fault
3.	option	fixed	switch for linker options
4.	error	label	error return
5.	code	fixed	error identifying code

The linker assumes that the fault occurred in a linkage section. It uses the machines conditions, arg1, to determine where the link pair exists. It then works through the linkage section pointers for the faulting reference. Linking is temporarily suspended if a trap before link has been requested. The linker ultimately constructs either an ITS or ITB pair in the link word pair and modifies the machine conditions so that the fault will not occur again when the machine is restored to its state before the fault.

Two options, specified by arg3, have presently been defined for the linker. If arg3 equals 0, as it does when the linker is called by f2catc, the full facilities in the

linkage section may be used. If arg3 equals 1, as it does when called by link_, restrictions are placed on the linker. It then ignores possible trap before link requests and does not allow type 2 external references. The need for these restrictions is described in the explanation of the link_ procedure. The linker calls the procedure segman to find the number of a segment. (see BE.8.03, Segment Management for the Pseudo-supervisor). If the segment is not listed in the segment name table, the linker calls the procedure search (see below) to determine whether the referenced segment exists in the segment library. If successful, search returns a segment number to the linker. After a successful return the segment will have been loaded along with a linkage section and a symbol table if either or both were present in the segment library.

The linker is used by the intersegment debugging aid (MSPM Section BE.12.01). If a process is using the debugging aid, the linker makes a call to the debugging procedure each time a linkage fault occurs.

The linker assumes that the first linkage block in a linkage section is at the beginning of a segment named LKnnnn, where nnnn is the octal segment number of the procedure or data segment that has linkage information.

Error codes for errors detected by the linker:

1. Fault occurred in linkage section with no link definitions.
2. Illegal external reference type code.
3. External symbol definition not found in linkage section.
4. Segment name not found in segment name table or in the segment library and trace procedure (see section BE.12.01) is not on.
5. Linkage section segment of external symbol not found in segment name table.
6. Tried to trap before linking with call ptr equal 0.
7. Arg3 value not defined.
8. Illegal control word returned to linker from trace procedure (see BE.12.01).
9. Illegal control word returned to linker from terror procedure (see BE.12.01).
10. Fault tag 2 encountered in a non-standard situation.

Search

The procedure to load segments from the segment library is called search. The segment, library_dictionary, is searched for the name of the referenced segment.

If the name is not found in the library_dictionary, the following error message is sent to the error file;

UNABLE TO FIND A LIBRARY ENTRY FOR THE FOLLOWING SEGMENT NAME (name of the segment referenced)

Search then returns to the linker and if the trace procedure (MSPM BE.12.01) is not in operation, execution is terminated. If the trace procedure is in effect, it is called. Upon returning to the linker from trace, the linker will continue to execute if a dummy link has been established, otherwise, it will be terminated.

When the segment name is found in the library_dictionary, search calls the pseudo-supervisor routine newseg to enter the segment into the segment name table. If there is an error, the message

UNABLE TO CREATE NEW TEXT SEGMENT FOR THE FOLLOWING SEGMENT NAME (name of the segment referenced)

is written on the error file. The procedure grow is called next to create space for the new segment. The error comment at this point is

UNABLE TO GROW NEW TEXT SEGMENT FOR THE FOLLOWING SEGMENT NAME (name of segment referenced).

If there is a linkage segment or a symbol table associated with the text segment being loaded, they are also created and space allocated. Error comments similar to the two above are made. Search finally calls the segment escape to do the actual loading of the routine(s). If there is an error at this point, the error comment is

ERROR IN LOADING FROM LIBRARY FOR THE FOLLOWING SEGMENT NAME (name of the segment referenced).

If any of the above errors is encountered, the run is terminated without returning to the linker. If no error is encountered, the segment number of the segment referenced is returned.

Forcing a link

A link may be set by calling

pseudo_supervisor$link_(point, error, code)

where the arguments are:

	identifier	attribute	meaning
1.	point	pointer	points to a link pair
2.	error	label	error return
3.	code	fixed	error identifying code

The procedure link_ is called with arg1 pointing to a link pair in some linkage section. If the link has not yet been set (i.e., if the word pointed to by arg1 contains an fi modifier), then the linker is called, and the link is set (i.e., replaced by an appropriate ITS pair). If the link is already set this procedure does nothing.

In the call to the linker, its argument 3 is set to equal 1. Therefore, the linker will ignore trap before link requests and not allow ITB type references. This option in the linker allows users to effectively eliminate traps in the linkage section without modifying the linkage information. ITB type addressing is not allowed since the appropriate base register values are not provided.

Error codes for errors in link_ are:
1-5. Same as linker error codes 1. -5.