TO: MSPM Distribution
FROM: M. A. Padlipsky
SUBJ: BE.7.04
DATE: 03/05/68

Note that this version of BE.7.04 does not supersede the 3/1/66 version of BE.7.04, which latter was renumbered BN.8 some time ago. That is, the attached section is BE.7.04, and "EPLBSA, Bootstrap Assembler for EPL" (3/1/66) should be BN.8, if it has not already been renumbered in your copy. To complicate matters further, BN.8 will be superseded eventually, by BX.5.00-.04; but at present it represents the best MSPM source of documentation of the language, per se.

Published: 03/05/68

Identification

EPLBSA in the GECOS Environment
J. D. Mills

Purpose

This section defines those interfaces which exist between EPLBSA and the GECOS operating system. Two kinds of information are included:

1. Usage - i.e. format of control cards and other points of information not part of the EPLBSA language but which might be needed in using EPLBSA.
2. System Interfaces - these are facilities of GECOS used by EPLBSA.

The information on the following subject is not included but can be obtained in the indicated places.

1. The EPLBSA language. See BX.5.00 - BX.5.04
2. The EPLBSA command in Multics. See BX.5.05
3. The internal organization of EPLBSA. See M0065 for a somewhat out of date discussion.

Control Cards

The library version of EPLBSA is located on the 645 library which is normally file code LB on permanent (PERM) storage. The name of the main program is EPLBSA and its entry is EPLBSA. The assembler uses core storage as buffer and working space and thus it requires about 64K of core. These requirements are met by the set of control cards listed below for a typical use of EPLBSA:

Col. 1	Col. 8	Col. 16
$	OPTION	NOMAP
$	LIBRARY	LB
$	USE	EPLBSA
$	ENTRY	EPLBSA

Col. 1	Col. 8	Col. 16	
$	EXECUTE	ON3	See Switch Settings
$	LIMITS	25, 64000,,10000	
$	DISC	LS,A1S,75L,EPLBSA.LISTING	
$	DISC	ST,A2S,10L,EPLBSA.SYMTB	
$	DISC	TX,A3S,30L,EPLBSA.TEXT	
$	DISC	LK,A4S,20L,EPLBSA.LINK	
$	DISC	ER,A5S,40L,ERROR.FILE	
$	TYPE	TY,A8R	
$	PERM	LB,EPLBSA	
$	DATA	9C	

The binary data cards follow this set of control cards. Normally this would all be done for the user via the MRGEDT command on CTSS and these cards would be, instead, card images on tape.

Switch Settings

As a point of further explanation, the $ EXECUTE card may have a number of optional switch settings other than ON3. The following possibilities exist:

ON1 - The assembler will give a snapshot dump of core when it finishes execution. However, DUMP must be requested.

ON2 - This turns off the on-line listing (equivalent to the pseudo-op NOLIST).

ON3 - This turns off the listing which is returned to CTSS via mrgedt.

ON4 - Unused.

ON5 - Unused.

ON6 - If on all dumps are purely octal. If off dumps are in extended form having BCD and op-code interpretations in addition to the octal.

DUMP - Must be specified to get a dump.

Input

The input to EPLBSA is a character stream without card boundaries or line marks. Characters are 7 bit ASCII code, imbedded in 9 bit sub-fields, four per 36 bit word. In reality, the assembler expects its input as column binary card images. Only words 3 through 24 of the card are read. The remainder are ignored. These card images are expected to be on file 9C which is normally a disc file if produced by 645 EPL or a data file if the input is from an IMCV tape or cards.

The GECOS routines used in input are: OPEN, CLOSE, REWND, and GET.

Output

The primary output of EPLBSA is a collection of three binary segments. They are written by EPLBSA into the three files TX (text), LK (link), and ST (symbol). As a convention, 24 word headers are written for each of these files. The first word of each of these headers is the CTSS file name, in six bit GE BCD, of the program being assembled. The second word is "TEXT", "LINK", or "SYMBOL". The next 22 words are all zeroes.

The output is buffered into 128 word blocks and then written with calls to PUT. The output files are, of course, opened (with calls to OPEN) before written. When assembly is completed the files are terminated with end-of-file markers written by WEF and then released by calls to RELSE. All output is written as variable-length logical records.

In addition to the binary segments EPLBSA produces two listing files, writes into the error file, and may type on the console typewriter.

The first listing is written onto system output file P* which then appears on the 645 printer. This listing is a GE BCD representation of the ASCII input. A title and page number are provided via the GECOS routines IOEDIT. The remaining routines used are OPEN, PRINT, and RELSE. Note that no end of file is written.

The second listing, which is written as an ASCII character stream into file LS, is provided for the return tape for eventual printing at a CTSS console. The routines used are OPEN, PUT (with variable-length logical records), WEF, and RELSE. An end-of-file is written by WEF before the file is released.

Short diagnostic messages are written into file ER. These messages are in ASCII and are written as variable-length records with PUT. Again OPEN and RELSE are used but no end-of-file is written.

Switch Retrieval

A sense-switch-like facility is provided in GECOS by allowing the user to set indicators on his $ EXECUTE card. These indicators are turned on by saying "ONi" where $1 \leq i \leq 6$. EPLBSA interrogates these switch settings by a call to the GECOS routine GERETS which returns a bit string of length 36 in the Q register. If the bits are numbered 0 to 35 going from left to right then bit 6 corresponds to ON1, bit 7 to ON2, etc.

EPLBSA interprets ON1, ON2, and ON3 as indicated under SWITCH SETTINGS. ON6 is interpreted by GECOS.

Free Storage

All internal tables and binary output are stored in list structures. The free storage for these lists is obtained in a very GECOS-dependent fashion. Figure 1 shows a diagram of how EPLBSA expects to be loaded.

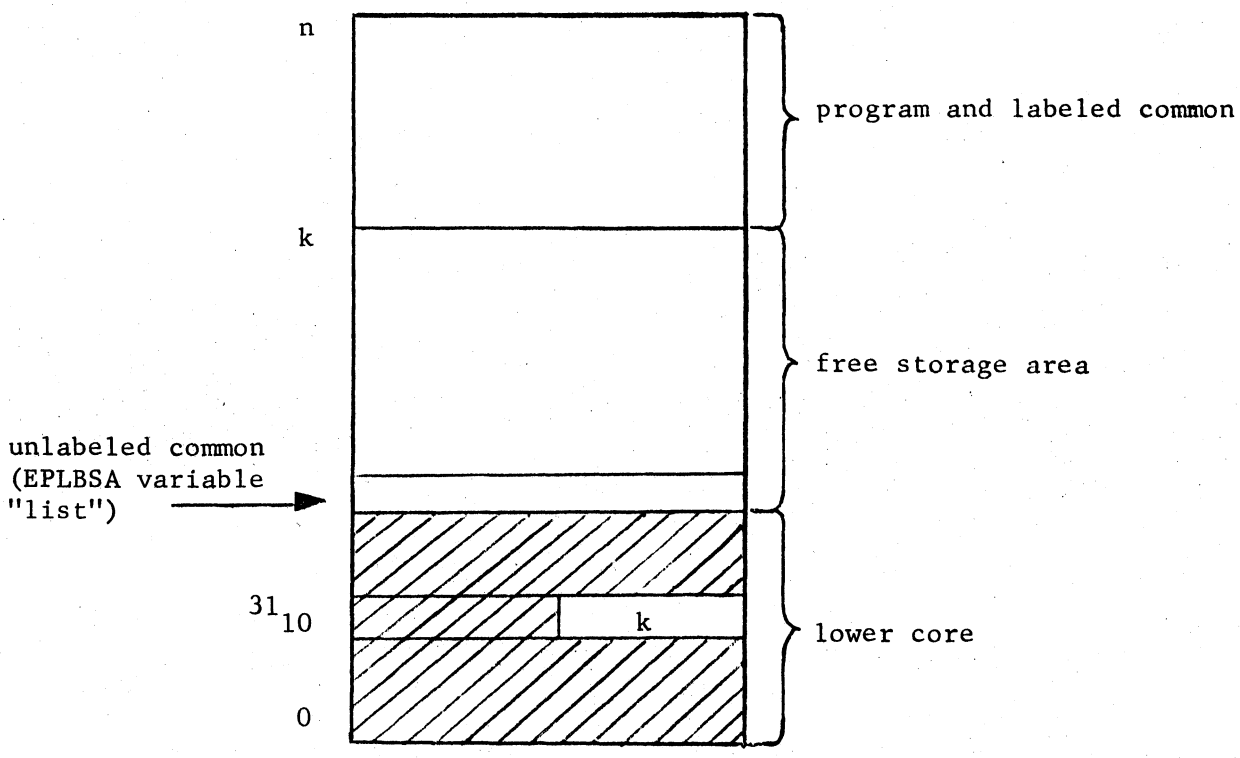

Figure 1: Loading diagram for EPLBSA.

Thus, EPLBSA knows that the contents of the right half of location 31(10) is the first available location below the loaded program and labeled common. Knowing also that the one-word unlabeled common variable "list" is assigned the location just above lower core, EPLBSA has the bounds for its free storage area. Any changes to these standards or the non-standard "lowload" option will seriously affect the execution of EPLBSA.

Termination

EPLBSA includes its own exit routines "EXIT" for normal termination and ".FEXIT" and ".FXEM." for abnormal termination.

In "EXIT" if "ON1" has not been set the terminate code "OK" is loaded into the Q register and then a MME GEFINI is executed. If "ON1" was set by the user then ".FEXIT" is called so that the user will get his requested dump.

In ".FEXIT" the unused portion of the free storage list is zeroed in preparation for a dump. Then GESNAP is invoked twice by a MME. GESNAP gives a snapshot dump of core memory of up to 32K locations. Since EPLBSA normally runs with 64K limits two GESNAP's are required.

After the dump "NG" is loaded into the Q register and a MME GEFINI is executed.