Computation Department

M-026

MASTER

LTSS Livermore Time-Sharing System

Part II: SERVICE PROGRAMS

Chapter 108: MONITOR 402 LOADER

Marilyn Richards, et al.

June 20, 1968 Edition - 1

LAWRENCE LIVERMORE LABORATORY
University of California

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

AUTHORS: HELEN GIN, MARILYN KARVELIS

PURPOSE: THIS CHAPTER OF 'LIVERMORE TIME SHARING SYSTEM' DESCRIBES THE USAGE OF THE MONITOR 402 LOADER.

HISTORY: CHAPTER 108 OF 'LIVERMORE TIME SHARING SYSTEM' SUPERCEDES THE LOADER DESCRIPTION IN CIC-MP-23, 'SO YOU WANT TO USE LRLTRAN.'

NOTICE

THIS REPORT IS ILLEGIBLE TO A DEGREE THAT PRECLUDES SATISFACTORY REPRODUCTION

IF YOU WISH TO BE PLACED ON A MAILING LIST FOR CORRECTIONS TO THIS CHAPTER OF LTSS, PLEASE SEND YOUR NAME AND L-CODE TO:

LTSS LIST
COMPUTER INFORMATION CENTER
L-60

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

NOTATION: IT IS FREQUENTLY NECESSARY TO SET WORDS AND PHRASES OFF FROM THE BODY OF THE TEXT. UNFORTUNATELY, SINCE THE ENTIRE TEXT IS WRITTEN IN CAPITAL LETTERS, WE ARE UNABLE TO USE CASE AS A DISTINGUISHING CHARACTERISTIC. THEREFORE, VARIOUS PUNCTUATION MARKS ARE USED.

A. LANGUAGE STATEMENTS AND MESSAGE STRINGS ARE SEPARATED FROM TEXT BY QUOTATION MARKS. FOR EXAMPLE: THE SYSTEM SENDS A MESSAGE 'PROBLEM ERROR 0200' TO THE TELETYPE ...

B. TO DEFINE A WORD OR PHRASE, THE TERM IS INCLUDED BETWEEN BRACKETS < .... >. FOR EXAMPLE: A <TELETYPE> IS A MACHINE ...

C. VARIOUS LEVELS OF NAMES ARE INCLUDED IN THE FOLLOWING TABLE:

LEVEL	PUNCTUATION	EXAMPLES
SYSTEM NAME	NONE	THE FROST SYSTEM...
FILE NAME	/..../	PUBLIC FILE /LRLLIB/ IS...
PROGRAM NAME	-....-	SUBROUTINE -ALTER- WILL...
VARIABLE NAME	'....'	SET 'ABC' EQUAL TO

D. IT IS ALSO FREQUENTLY NECESSARY TO INDICATE CONTENTS AND LOCATIONS. THIS IS DONE BY THE FOLLOWING:

	PUNCTUATION	EXAMPLES
CONTENTS OF	[....]	[ABC] = 4.
LOCATION OF	(....)	(ABC) = 10472
		(LRLLIB)= DISC ADDRESSES (107-1123)

THIS NOTATION ALLOWS US TO DISTINGUISH BETWEEN DIFFERENT ENTITIES THAT HAVE THE SAME NAMES. FOR EXAMPLE, 'PUBLIC FILE /OUT/ CONTAINS PROGRAM -OUT-'. PUNCTUATION IS OFTEN OMITTED IF THE TERM IS ON A LINE OF ITS OWN, IS IN A TABLE, OR IS IN A LIST.

CROSS REFERENCES:

1. CROSS REFERENCES WITHIN CHAPTER 108 ARE INDICATED BY CHAPTER.SECTION.SUBSECTION AND PAGE NUMBER: (SEE 108.7.1, PAGE 20).

2. CROSS REFERENCES TO OTHER CHAPTERS ARE INDICATED IN THE SAME WAY, BUT WITHOUT PAGE NUMBERS: (SEE 4.7.3 AND 5.8).

-TABLE OF CONTENTS- PAGE

108.1. INTRODUCTION

THE 6600 MONITOR402 LOADER MAKES POSSIBLE SEPARATE COMPILATIONS OF SUBROUTINES CONSTITUTING AN ENTIRE PROGRAM. EACH SUBROUTINE IS COMPILED AS IF IT WERE THE ONLY ONE IN A PROGRAM. THAT IS, ALL ADDRESSES ARE RELATIVE TO THE START OF THE SUBROUTINE COMPILED OR ASSEMBLED. ADDRESSES NOT DEFINED IN THE SUBROUTINE (EXTERNAL SYMBOLS) AND THOSE DEFINED BUT NOT STORED WITHIN THE SUBROUTINE (COMMON BLOCKS) ARE FLAGGED AS SUCH BY THE COMPILER (I.E., FORTRAN, PL1) OR THE ASSEMBLER (I.E., RVIVID, FORTRAN ASSEMBLER).

THE LOADER DECIDES WHERE IN MEMORY THE SEPARATE SUBROUTINES WILL BE LOCATED *(1)* AND DEFINES THE ADDRESSES REFERENCED BY THE SUBROUTINE WHICH WERE NOT KNOWN AT THE COMPILE/ASSEMBLE PHASE. IN ADDITION, ALL ADDRESSES RELATIVE TO THE START OF A GIVEN SUBROUTINE, AND THEREFORE KNOWN AT COMPILE/ASSEMBLE TIME, WILL BE RELOCATED BY THE LOADER (I.E., WILL BE INCREMENTED BY THE ABSOLUTE FIRST WORD ADDRESS COMPUTED BY THE LOADER). THE ABSOLUTE FIRST WORD ADDRESS IS WHERE THE BEGINNING OF THE GIVEN SUBROUTINE IS PLACED IN MEMORY RELATIVE TO THE ENTIRE PROGRAM.

THE EXTERNAL SYMBOLS MAY BE OF SEVERAL TYPES:

1. OTHER SUBROUTINES
2. EXTERNALLY DEFINED PARAMETER VALUES *(2)*
3. SEGMENTS WITHIN THIS SUBROUTINE OR ANOTHER SUBROUTINE

THE COMMON ADDRESSES MAY BE:

1. LABELED COMMON BLOCKS, OR
2. BLANK/NUMBERED COMMON BLOCKS. *(3)*

IF THE FIRST CHARACTER OF THE NAME OF A COMMON BLOCK IS ALPHABETIC IT IS A LABELED COMMON BLOCK. IF THE FIRST CHARACTER IS NUMERIC IT IS A NUMBERED COMMON BLOCK. BLANK IS THE ABSENCE OF A NAME AND IS PROCESSED AS IF IT WERE A NUMBERED COMMON BLOCK. THE LABELED COMMON BLOCKS ARE CONSIDERED (FOR PURPOSES OF MEMORY ALLOCATION) AS SUBROUTINES AND ARE INTERSPERSED AMONG THEM. THE NUMBERED BLOCKS ARE GROUPED TOGETHER, SEPARATELY FROM THE SUBROUTINES.

----

(1) SEE 108.7. ON PAGE 28.
(2) SEE 108.8.5. ON PAGE 30.
(3) SEE 108.8.4. ON PAGE 30.

108.1. INTRODUCTION

THE FIRST DEFINITION OF A SUBROUTINE, OF A COMMON BLOCK, OF A PARAMETER VALUE, OR OF A SEGMENT IS THE ONE THE LOADER WILL USE AND SUBSEQUENT DIFFERENT DEFINITIONS RESULT IN A COMMENT TO THE USER.

EXAMPLE 1:

IF TWO SUBROUTINES HAVE THE SAME NAME ONLY THE FIRST ONE WILL BE LOADED AND THE SECOND ONE WILL BE SKIPPED AND RESULT IN A 'MULTIPLY DEFINED' COMMENT.

EXAMPLE 2:

IF A PARTICULAR COMMON BLOCK IS FIRST DEFINED AS HAVING A LENGTH X AND LATER IS DEFINED AS HAVING LENGTH X+Y, THE FIRST LENGTH WILL BE USED AND THE SECOND WILL CAUSE AN 'INCONSISTENT LENGTH' MESSAGE. IF THE LATER LENGTH IS X-Y, X WILL BE USED AND NO MESSAGE OCCURS.

THE LINKING TOGETHER OF INDEPENDENTLY COMPILED SUBROUTINES IS ACCOMPLISHED BY THE LOADER INSERTING THE ADDRESS (WHICH ONLY THE LOADER KNOWS) OF THE CALLED ROUTINE INTO THE WORD GENERATED BY THE COMPILER/ASSEMBLER. IF A SUBROUTINE IS NEEDED BY A PROGRAM AND IS NOT SUPPLIED BY THE PROGRAM, THE LOADER WILL SEARCH LIBRARY FILES TO LOCATE THE MISSING SUBROUTINE. IF THE USER SUPPLIES A PERSONAL LIBRARY FILE (SEE CHAPTER 301), THIS LIBRARY WILL BE SEARCHED FIRST. IF IT CANNOT BE FOUND THERE, A LIBRARY FILE SUPPLIED BY THE SYSTEM WILL BE SEARCHED NEXT. IF THE SYSTEM LIBRARY DOES NOT CONTAIN THE MISSING SUBROUTINE, THE LOADER WILL DO ONE OF TWO THINGS:

1. TERMINATE THE JOB, OR
2. IF A /* DEBUG/ CARD IS SPECIFIED, CONTINUE THE LOADING PROCESS AND LINK THE CALL TO THE MISSING SUBROUTINE TO A STANDARD SUBROUTINE ON THE SYSTEM LIBRARY FILE. THIS STANDARD SUBROUTINE (OCOMIS) WILL PRINT AN APPROPRIATE MESSAGE WHEN CALLED DURING EXECUTION AND TERMINATE THE EXECUTION.

IN ANY CASE, A LIST OF ALL MISSING SUBROUTINES WILL BE GIVEN TO THE USER PRIOR TO THE END OF THE LOAD PROCESS.

108.1. INTRODUCTION

A MORE COMPLEX LEVEL OF LINKING (OTHER THAN THE SIMPLE CALL JUST DESCRIBED) IS AFFORDED BY THE USE AND EVALUATION OF THE INFORMATION ON THE -RAX- AND -LIT- CARDS. *(4)* THE -RAX- CARD HAS THE ABILITY TO DESCRIBE AN ARITHMETIC EXPRESSION INVOLVING THE ADDRESS OF AN EXTERNAL SYMBOL (I.E., THE ADDRESS OF ANOTHER SUBROUTINE). THIS EXPRESSION CAN CONTAIN CONSTANTS AND THE FOLLOWING ARITHMETIC OPERATORS:

+ ADDITION
- UNARY MINUS AND BINARY MINUS
* MULTIPLICATION
/ DIVISION

THERE ARE TWO OTHER OPERATORS WHICH DEFINE WHETHER THE COMPUTED EXPRESSION IS A FULL WORD OR THE ADDRESS PORTION OF A WORD. IN ADDITION THERE EXIST THREE OPERATORS FOR USE IN BIT FIELD RELOCATION. THEY DETERMINE THE WIDTH OF THE BIT FIELD (I.E., NUMBER OF BITS), THE OFFSET OF THE BIT FIELD (I.E., WHERE IN THE WORD THE FIELD STARTS), AND THE BEGINNING OF A DIFFERENT BIT FIELD WITHIN THE SAME COMPUTER WORD. *(5)*

THE OPERANDS IN THE ARITHMETIC EXPRESSION DEFINED ON THIS CARD ARE THE EXTERNAL SYMBOL NAMES DESCRIBED BY THE -EXT- CARD *(6)* AND/OR THE CONSTANTS DEFINED ON THE -LIT- CARD. THE CONSTANTS DEFINED ON THE -LIT- CARD CAN BE ABSOLUTE OR RELATIVE TO THE START OF ANY COMMON BLOCK, THE GIVEN SUBROUTINE FIRST WORD ADDRESS, OR A SEGMENT *(7)* WITHIN THE SUBROUTINE.

----

(4) SEE 108.8.7. ON PAGE 30 AND 108.8.8. ON PAGE 31 FOR DESCRIPTIONS OF THESE CARDS.

(5) SEE THE TABLE OF OPERATORS IN 108.8.8. ON PAGE 31.

(6) SEE 108.8.6. ON PAGE 30 FOR A DESCRIPTION OF THIS CARD.

(7) SEE 108.8.2. ON PAGE 29.

108.2. GENERAL LOAD TYPES

108.2.1. CHAIN VERSUS NON-CHAIN

NON-CHAIN: *(8)*

THIS IS THE MOST USED OF THE GENERAL LOAD TYPES. IT IS APPLICABLE FOR SMALL TO MEDIUM-LARGE SIZE PROGRAMS (I.E., PROGRAMS WHOSE TOTAL SIZE WILL BE CONTAINED IN ONE CORE LOAD). THIS TYPE IS LOADED IN ONE PIECE AND EXECUTED IN A STRAIGHT-FORWARD MANNER.

CHAIN: *(8)*

A CHAIN TYPE LOAD IS NECESSARY IF THE TOTAL SIZE OF A PROGRAM EXCEEDS THE SIZE OF THE MACHINE MEMORY OR IF THE USER WISHES TO REDUCE THE AMOUNT OF SPACE USED DURING EXECUTION. A CHAIN LOAD, THEN, CONSISTS OF MORE THAN ONE CORE-LOAD OR CORE-IMAGE. THESE DIFFERENT CORE IMAGES HAVE A UNIQUE IDENTITY AND HENCE CAN BE REFERENCED (I.E., CALLED INTO CORE) DURING EXECUTION.

TO START EXECUTION, A PARTICULAR CORE-IMAGE (TERMED A LINK) IS SPECIFIED TO BE LOADED INTO THE MACHINE FOR EXECUTION. WHEN THE PROGRAM HAS FINISHED ALL OF THE NECESSARY EXECUTION FOR THIS PARTICULAR CORE-IMAGE, A CALL IS MADE TO THE SYSTEM (I.E., MONITOR SYSTEM, VIA THE -CHAIN- ROUTINE) TO READ ANOTHER CORE-IMAGE INTO THE MACHINE. THIS CORE-IMAGE MAY PARTIALLY OR TOTALLY OVERLAP THE ONE PREVIOUSLY EXECUTING OR IT MAY BE ADJACENT TO IT. EXECUTION OF THIS NEW CORE-IMAGE (LINK) IS STARTED AT A POINT SPECIFIED AT LOAD TIME. NOT ALL LINKS IN A GIVEN CHAIN PROGRAM NEED BE EXECUTED, AND CONVERSELY A PARTICULAR LINK MAY BE EXECUTED MANY TIMES, NOT NECESSARILY SEQUENTIALLY.

COMMUNICATION BETWEEN THE UNIQUE CORE LOADS IS ACCOMPLISHED THROUGH AN AREA COMMON TO ALL OF THEM. THIS AREA IS DESIGNATED AS THE BLANK/NUMBERED COMMON AREA. *(9)* IT IS THE USER'S RESPONSIBILITY TO INITIALIZE THIS AREA OF THE PROGRAM. ONCE EXECUTION IS BEGUN, THIS AREA IS SACRED (I.E., IT WILL NOT BE WRITTEN OVER BY ANY PART OF THE SYSTEM). THE BEGINNING (I.E., THE FIRST WORD ADDRESS) OF THIS AREA IS THE SAME FOR ALL LINKS IN A CHAIN PROGRAM AND THIS FEATURE IS THE ONLY THING THAT DEFINES IT AS A COMMON COMMUNICATION AREA. DIFFERENT LINKS MAY, IF THERE IS A NEED TO, STRUCTURE THE AREA DIFFERENTLY (I.E., THE STRUCTURE OF THE COMMON AREA IS NOT COMMON TO ALL LINKS, ONLY THE FIRST WORD ADDRESS IS). *(10)*

----

(8) SEE THE LOAD TYPE EXAMPLE IN 108.5.1. ON PAGE 17 FOR MEMORY ALLOCATION.
(9) SEE 108.8.4. ON PAGE 30.
(10) SEE 108.3. ON PAGE 9 FOR A DISCUSSION OF DIFFERENT CHAIN LOADS.

108.2. GENERAL LOAD TYPES

108.2.2. CONTROLLEE VERSUS NON-CONTROLLEE

NON-CONTROLLEE:

A NON-CONTROLLEE PROGRAM EXECUTES DIRECTLY UNDER THE CONTROL OF THE MONITOR SYSTEM. THIS MAKES POSSIBLE VARIOUS AUTOMATIC DEBUGGING AIDS SUCH AS MEMORY DUMPS, EMPTYING OF HSP OUTPUT AND PRINTER BUFFERS, ETC. ALSO, ALL OUTPUT FILES ARE DIRECTED TO THOSE SECTIONS OF THE FROST OPERATING SYSTEM FOR PROPER DISPERSAL.

CONTROLLEE:

CONTROLLEE PROGRAMS DO NOT EXECUTE UNDER THE CONTROL OF THE MONITOR SYSTEM. THIS LOAD TYPE WAS CREATED TO ALLOW THE USER TO CONTROL HIS OWN EXECUTION DIRECTLY FROM THE TELETYPE OR UNDER CONTROL OF SOME OTHER CONTROLLEE. SOME CHANGES SEEMED DESIRABLE FROM THE USER'S VIEW IF CONTROL WAS INITIATED BY HIS TELETYPE. ONE OF THESE IS THE AUTOMATIC CHANGING, BY THE LOADER, OF THE HARDWARE TYPE FOR PRINTER I/O COMMANDS. ALL PRINTER OUTPUT IS DIRECTED TO THE TELETYPE. CALLS TO THE MONITOR SYSTEM ROUTINE -EXIT- ARE REDIRECTED TO THE ROUTINE -QUIT-. THE FORMER ROUTINE SENDS A MESSAGE TO THE CONTROLLER (ASSUMED TO BE THE PUBLIC FILE /MONITOR/) WHICH IS THEN TO TAKE SPECIFIC ACTION. THE -QUIT- ROUTINE, ON THE OTHER HAND, MAKES A FROST SYSTEM CALL TELLING THE OPERATING SYSTEM (FROST) IT WISHES TO TERMINATE. IT IS THEN UP TO THE USER TO DO AS HE WISHES FROM THEN ON: FILE CLEANUP, ETC.

IF THE USER WISHES A CONTROLLEE LOAD, HE ALSO HAS THE OPTION OF SPECIFYING THE ENTRY POINT TO BEGIN EXECUTION (THIS IS STORED IN THE EXCHANGE JUMP PACKAGE IN THE CONTROLLEE FILE PRECEDING THE INSTRUCTIONS). BY DEFAULT THIS IS -MAIN.-.

CONTROLLEE/NON-CONTROLLEE:

IF A PROGRAM IS WRITTEN CORRECTLY (NAMELY IN THE AREA OF POSSIBLE TELETYPE COMMUNICATION AND INTERACTION) IT CAN BE RUN EITHER AS A CONTROLLEE OR NON-CONTROLLEE. THIS HAS ITS ADVANTAGES IN THE FACT THAT NON-CONTROLLEES WILL BE EXECUTED BY MONITOR IN OFF-HOURS THROUGH A 'BATCH PROCESS' TECHNIQUE (WHICH IS THE ONLY MODE OF MONITOR EXECUTION ON OTHER MACHINES IN THE LABORATORY). BY RUNNING IT AS A CONTROLLEE DURING THE DAYTIME HOURS, ONE CAN INTERACT WITH THE PROGRAM TO UTILIZE TIME SHARING FEATURES. THE CODE SHOULD BE WRITTEN IN SUCH A MANNER AS TO ACCEPT INPUT FROM A TAPE OR DISC FILE (NON-CONTROLLEE) OR TELETYPE (CONTROLLEE), IN OTHER WORDS, IT SHOULD BE ABLE TO SET A FLAG INDICATING IN WHICH MODE IT IS EXECUTING.

108.3. CHAIN LOADS

108.3.1. CHAIN FILE FORMAT

THE SEPARATE LINKS OF A CHAIN LOAD EACH CONSTITUTE ONE LOGICAL RECORD ON THE CHAIN FILE (TAPE OR DISC FILE). THESE RECORDS ARE COMPRISED OF THE INSTRUCTIONS, PRIVATE DATA STORAGE (I.E., BSS), AND LABELED COMMON BLOCKS. NOTE THAT BLANK AND NUMBERED COMMON BLOCKS ARE NOT PART OF THE LINK RECORD, HENCE THE REASON FOR NOT HAVING THE CAPABILITY OF PRE-LOADING THESE COMMON BLOCKS. THE GREAT MAJORITY OF CHAIN LOADS DECLARE VERY LARGE BLANK AND NUMBERED BLOCK STORAGE, SO THAT THIS PROCEDURE GREATLY REDUCES THE SIZE OF THE CHAIN FILE AND REDUCES THE AMOUNT OF TIME REQUIRED TO READ THESE LINKS INTO MEMORY. IT ALSO ELIMINATES THE NEED TO READ OUT AND WRITE IN THESE LARGE AREAS BETWEEN COMMANDS TO BRING IN ANOTHER LINK FOR EXECUTION.

HEADING THE CHAIN FILE IS A MAP OF THE FILE. THIS MAP CONSISTS OF TWO WORDS FOR EACH LINK IN THE FILE. THE ORDER OF THE WORDS IS THE SAME AS THE PHYSICAL ORDER OF THE LINKS IN THE FILE. THE MAP CONTAINS THE FOLLOWING INFORMATION: RELATIVE FIRST WORD ADDRESS OF THE LINK, LENGTH OF THE LINK, MEMORY REQUIRED TO EXECUTE THE LINK, LINK NUMBER (FIRST ARGUMENT ON /* CHAIN/ CONTROL CARD), LOGICAL UNIT NUMBER OF THE FILE (SECOND ARGUMENT ON THE /* CHAIN/ CONTROL CARD), RELATIVE DISC ADDRESS OF THE BEGINNING OF THE LINK (THIS IS IGNORED IF THE FILE IS A MAGNETIC TAPE), AND THE TRANSFER ADDRESS TO BEGIN EXECUTION.

108.3.2. CONVENTIONAL CHAIN LOADS

IN A CONVENTIONAL CHAIN LOAD, ALL OF THE SEPARATE LINKS (CORE-IMAGES) ARE OF THE SAME LEVEL. THAT IS, EACH LINK AS IT IS WRITTEN INTO MEMORY FOR EXECUTION ALWAYS OVERLAPS THE PREVIOUSLY EXECUTING LINK. THUS, IF A GIVEN LINK IS TO BE EXECUTED BEFORE AND AFTER ANY OTHER LINK, IT NEEDS TO BE RE-WRITTEN INTO MEMORY. *(11)*

THE CONTROL CARDS IN THE INPUT FILE FOR THIS TYPE OF LOAD ARE /* CHAIN (L,T)/ CARDS. THESE LINKS DO NOT SHARE ANY LABELED COMMON BLOCKS OR INSTRUCTION SEQUENCES. THE ONLY THING THEY HAVE IN COMMON IS THE FIRST WORD ADDRESS OF THE BLANK/NUMBERED COMMON AREA.

----

(11) SEE 108.5.6. ON PAGE 22 AND 108.5.7. ON PAGE 22 FOR EXAMPLES OF MEMORY LAYOUTS.

108.3.3. MAIN-OVERLAY-SEGMENT CHAIN LOADS

THE CONTROL CARDS REQUIRED FOR THIS LOAD ARE /* MAIN (L,T)/, /* OVERLAY (L1,T,ENTRYNAME)/, AND /* SEGMENT (L2,T,ENTRYNAME)/. NOTE: /* MAIN (L,T)/ AND /* CHAIN (L,T)/ ARE INTERCHANGEABLE AND DEFINE EXACTLY THE SAME TYPE OF LINK. THE ENTRY NAME ON THE ABOVE CONTROL CARDS SPECIFIES THE ENTRY POINT NAME TO WHICH THE USER WISHES TO SEND INITIAL CONTROL AFTER THE LINK HAS BEEN READ INTO MEMORY.

THE LAST WORD ADDRESS OF EACH OVERLAY LINK LINKED TO THE SAME MAIN LINK (I.E., ALL /* OVERLAY/ CONTROL CARDS FOLLOWING A /* MAIN/ CONTROL CARD WITHOUT AN INTERVENING /* MAIN/ CONTROL CARD) IMMEDIATELY PRECEDES THE THE FIRST WORD ADDRESS OF THE MAIN LINK AND IS THE SAME FOR EACH OVERLAY. THE SAME RELATIONSHIP EXISTS BETWEEN OVERLAY LINKS AND SEGMENT LINKS. THE ONLY EXECPTION TO THIS IS THE ZEROLOAD-CONTROLLEE M-O-S CHAIN LOAD. *(12)*

NOTE: A SEGMENT LINK CANNOT BE DEFINED UNLESS AN OVERLAY LINK HAS BEEN PREVIOUSLY DEFINED. HOWEVER, THE /* OVERLAY/ CARD WITHOUT A PRECEDING /* MAIN/ OR /* CHAIN/ CARD WILL SERVE TO ACT AS A /* MAIN/ OR /* CHAIN/ CARD. IN THIS LATTER INSTANCE THE RESIDENT PORTION OF MONITOR TAKES THE PLACE OF A MAIN LINK. IN ALL OTHER NON-ZEROLOAD CONTROLLEE CASES, THE RESIDENT PORTION OF MONITOR BECOMES AN EXTENSION OF EACH AND EVERY MAIN LINK. IF THE USER WANTS A ZEROLOAD-CONTROLLEE MAIN-OVERLAY-SEGMENT LOAD, *(12)* THE /* MAIN/ CARD MUST APPEAR.

ALL ENTRY POINTS AND LABELED COMMON BLOCKS IN A MAIN LINK CAN BE LINKED TO ALL OF ITS ASSOCIATED OVERLAY AND SEGMENT LINKS (I.E., DIRECT CALLS TO ENTRY POINTS AND DIRECT REFERENCES TO LABELED COMMON ARE ALLOWED). THE SAME IS TRUE OF AN OVERLAY LINK AND ITS ASSOCIATED SEGMENT LINKS. HOWEVER, THE ONLY LINK THAT IS CAPABLE OF PRE-LOADING A GIVEN LABELED COMMON BLOCK IS THE FIRST LINK THAT MENTIONS THE BLOCK NAME (I.E., THE LINK THAT DEFINES THE BLOCK).

THE SUBROUTINE LINKAGE DOES NOT HOLD TRUE IN THE OPPOSITE DIRECTION. THE MAIN LINK CANNOT DIRECTLY CALL AN ENTRY POINT DEFINED IN AN ASSOCIATED OVERLAY OR SEGMENT LINK NOR CAN AN OVERLAY LINK DIRECTLY CALL AN ENTRY POINT DEFINED IN AN ASSOCIATED SEGMENT LINK.

ALL LINKS IN THIS TYPE CHAIN LOAD HAVE IN COMMON THE FIRST WORD ADDRESS OF THE BLANK AND NUMBERED COMMON BLOCK AREA.

----

(12) SEE 108.5.10. ON PAGE 25.

108.3. CHAIN LOADS

FOR ALL NON-ZEROLOAD CONTROLLEE CHAIN LOADS, *(13)* THE FIRST WORD ADDRESS OF THE BLANK/NUMBERED COMMON BLOCK AREA FOLLOWS IMMEDIATELY AFTER (I.E., THE NEXT HIGHER ORDER ADDRESS) THE RESIDENT PORTION OF MONITOR. SUBSEQUENT BLOCKS IN THIS AREA ARE ASSIGNED STORAGE ADDRESSES IN NUMERICALLY ASCENDING ORDER. THE FIRST WORDS OF INSTRUCTION BLOCKS (I.E., SUBROUTINES) AND OF LABELED COMMON BLOCKS WITHIN THE INDIVIDUAL LINKS ARE ASSIGNED STORAGE ADDRESSES IN NUMERICALLY DESCENDING ORDER.

THEREFORE IN CONVENTIONAL M-O-S CHAIN LOADS THE FIRST WORD ADDRESS OF EACH LOWEST LEVEL LOAD (SEGMENT IS THE LOWEST LEVEL) IMMEDIATELY FOLLOWS (I.E., NEXT HIGHER ORDER ADDRESS) THE LAST WORD OF THE BLANK/NUMBERED COMMON AREA DEFINED BY THAT LINK AND ITS ASSOCIATED HIGHER LEVEL LINKS (WHICH ARE AT MOST TWO, ITS OVERLAY LINK AND THE OVERLAY LINK'S MAIN LINK). THE LENGTH OF THIS BLANK/NUMBERED BLOCK IS THE LONGEST SUCH BLOCK DEFINED BY ANY ONE OF THE THREE POSSIBLE LINKS INVOLVED. *(14)*

IN ZEROLOAD-CONTROLLEE MAIN-OVERLAY-SEGMENT LOADS, THE LINKAGE RULES ABOVE APPLY. HOWEVER, THE MEMORY LAYOUT IS IN REVERSE ORDER. *(15)*

----

(13) SEE 108.5.7. ON PAGE 22 AND 108.5.9. ON PAGE 24.
(14) SEE 108.5.8. ON PAGE 23 AND 108.5.9. ON PAGE 24 FOR EXAMPLES OF MEMORY LAYOUT.
(15) SEE 108.5.10. ON PAGE 25 FOR AN EXAMPLE OF MEMORY LAYOUT.

108.4. MEMORY MAP AND MEMORY DUMP

MEMORY MAPS, PRODUCED AND PRINTED BY THE LOADER AT THE END OF THE LOADING PHASE, ARE MAINLY USEFUL FOR DEBUGGING PURPOSES TO ENABLE THE USER TO READ MEMORY DUMPS. THERE ARE THREE SECTIONS IN A MEMORY MAP. THE FIRST ARE THE SUBROUTINE NAMES, SECOND ARE THE ENTRY POINT NAMES, AND LAST ARE THE COMMON BLOCK NAMES.

108.4.1. SUBROUTINE NAMES

THESE NAMES APPEAR IN THE MAP IN THE SAME ORDER AS THE LOADER ENCOUNTERED THEM IN THE INPUT DECK. FOR NON-CHAIN JOBS THE FIRST ONE IS THE LEFT-MOST NAME ON THE FIRST LINE. FOR CONVENTIONAL CHAIN JOBS, THE FIRST ONE IS THE RIGHT-MOST NAME ON THE LAST LINE. FOR ALL JOBS, THE RELOCATED ADDRESSES ASSOCIATED WITH EACH NAME ARE IN NUMERICALLY ASCENDING ORDER. THESE RELOCATED ADDRESSES CORRESPOND TO THE FIRST WORD ADDRESS (RELATIVE WORD ZERO) OF THE SUBROUTINE NAME IMMEDIATELY TO THE LEFT OF THEM. THESE ADDRESSES ARE THE ABSOLUTE LOCATION OF THE WORD ZERO (SEEN IN THE OCTAL LISTING FOLLOWING THE SOURCE LISTING) OF THE SUBROUTINE IN THE MEMORY DUMP.

EXAMPLE 1

```
    MEMORY MAP:   NAME      LOCATION
                  SUBA        020152
    OCTAL LISTING OF -SUBA- (AN EXAMPLE FROM /LRLTRAN/ COMPILER OCTAL LISTING):
         000000 0400000003---------- P1          JBE 0,...
                ----------46000-----             PASS
                ---------------46000             PASS
    MEMORY DUMP:
         LOCATION 020152  0400020155460004600O
```

NOTE: THE 'P1' ON THE FIRST LINE OF OCTAL LISTING DENOTES THAT THE ADDRESS (00003) PORTION OF THE FIRST INSTRUCTION IS PROGRAM RELOCATABLE AND THE RELOCATION BASIS IS ONE. THIS MEANS THAT EVERY TIME THE LOADER SEES A RELOCATION BASIS=1 (SEE DESCRIPTION OF -RBD- CARDS) IT WILL ADD TO THE RELATIVE ADDRESS (00003) THE VALUE '20152,' I.E., '20152' IS THE VALUE ASSOCIATED WITH RELOCATION BASIS 1, FOR THE SUBROUTINE -SUBA-.

EXAMPLE 2

```
MEMORY MAP:   NAME      LOCATION
              SUBA       020152
OCTAL LISTING:
     000103 00474556456241645662            BCD GENERATOR
MEMORY DUMP:
     LOCATION 020255  00474556456241645662
```

NOTE: THERE WAS NO RELOCATION BASIS ASSOCIATED WITH THE CONTENTS OF RELATIVE WORD 103 OF SUBROUTINE -SUBA-. THEREFORE THE LOADER WILL NOT ALTER THE CONTENTS OF THAT LOCATION. THE LOCATIONS OF -SUBA-, NOT TO BE CONFUSED WITH THE CONTENTS OF THE LOCATION, HAVE A RELOCATION BASIS OF 1, WHICH HAS THE VALUE '20152' AS SEEN ABOVE IN THE MAP. HENCE THE ADDITION OF '000000+20152=20152' IN EXAMPLE 1 AND '000103+20152=20255' IN EXAMPLE 2 TO LOCATE THESE RELATIVE WORDS IN THE MEMORY DUMP.

108.4.2. ENTRY POINT NAMES

THESE NAMES WITH THEIR ASSOCIATED RELOCATED ADDRESSES ARE SORTED INTO ALPHABETICAL ORDER PRIOR TO PRINTING IN THE MAP. THESE DO NOT NECESSARILY HAVE A ONE-TO-ONE CORRESPONDENCE WITH THE SUBROUTINE NAME PORTION OF THE MAP SINCE ONE SUBROUTINE MAY HAVE MULTIPLE ENTRY POINTS.

EXAMPLE 3

```
MEMORY MAP:   NAME      LOCATION
              SUBA       020153
OCTAL LISTING OF -SUBA-:
     000001 0200000001---------- P1          SUBA JMP,SUBA
            ----------46000-----             PASS
            ---------------46000             PASS
     000002 5170000001---------- P1          STI 7...
            ----------5170000150 P1          STI 7...
MEMORY DUMP:
     LOCATION 020153  02000201534600046000
              020154  51700201535170020324
```

108.4. MEMORY MAP AND MEMORY DUMP

WHEN A CALL TO SUBROUTINE -SUBA-, I.E., 'CALL SUBA', IS MADE FROM A SUBROUTINE -SUBB-, THE BCD NAME 'SUBA' APPEARS AS AN EXTERNAL SYMBOL WITHIN -SUBB-.

EXAMPLE 4

OCTAL LISTING OF -SUBB- CORRESPONDING TO FORTRAN STATEMENT 'CALL SUBA':

```
000077 XXXXXXXXXX----------
       ----------7170000101 P1          FI 7,0,$C.1+0
000100 7160000000----------             FI 6,0,(0)+0
       ----------0200777776 X21         J,0,0,SUBA+1
           .
           .
           .
EXTERNAL SYMBOLS
SUBA
```

108.4. MEMORY MAP AND MEMORY DUMP

NOTE: THE 'X' IN RELATIVE WORD 100 (OCTAL) DENOTES EXTERNAL SYMBOL RELOCATION AND THE RELOCATION BASIS 21 IS THE SAME (I.E., 21) FOR ALL EXTERNAL SYMBOLS WITHIN -SUBB-, IN THIS EXAMPLE. THE ADDRESS PORTION (777776) IS ASSOCIATED WITH AN IDENTICAL (77776) POINTER ON A -RAX- CARD IN THE BINARY DECK FOR -SUBB-. THE -RAX- ENTRY CONTAINS A REFERENCE TO THE EXTERNAL SYMBOL CARD (-EXT-). ON THE -EXT- CARD SOMEWHERE APPEARS THE BCD NAME 'SUBA'. THE -RAX- ENTRY INSTRUCTS THE LOADER TO ADD THE LITERAL 1 TO THE VALUE OF THE EXTERNAL SYMBOL 'SUBA'. THE VALUE OF THE EXTERNAL SYMBOL 'SUBA' IS THE RELOCATED ADDRESS OF THE ENTRY POINT NAME 'SUBA' (NOT THE SUBROUTINE NAME -SUBA-). THEREFORE:

```
    EXAMPLE 5
         MEMORY MAP:   ENTRY POINT NAMES
                       NAME      LOCATION
                       SUBA        020153

                       SUBROUTINE NAMES
                       NAME      LOCATION     NAME      LOCATION
                       SUBA        020152     SUBB        030332

         MEMORY DUMP (LOCATION 30431 CORRESPONDING TO RELATIVE LOCATION 77 IN  -SUBB-  AND
         LOCATION 20154 CORRESPONDING TO RELATIVE LOCATION 2 IN -SUBA-):

              LOCATION 20154  51700201535170020324 (SAME AS IN EXAMPLE 3 ABOVE)
                              .
                              .
                              .
                       30431  ----------7170030433
                       30432  71600000000200020154
```

EXAMPLES 4 AND 5 EXPLAIN HOW THE LOADER ACCOMPLISHES SUBROUTINE LINKAGE.

108.4. MEMORY MAP AND MEMORY DUMP

108.4.3. COMMON BLOCK NAMES

THESE ENTRIES CONSIST OF THREE PARTS (NAME, LOCATION AND LENGTH) AS OPPOSED TO TWO PARTS (NAME AND LOCATION) FOR SUBROUTINE NAMES AND ENTRY POINT NAMES. THE ENTRIES ARE SORTED ACCORDING TO NUMERICALLY INCREASING RELOCATED ADDRESSES (I.E., LOCATION).

WITHIN A GIVEN SUBROUTINE, THE RELOCATION BASIS FOR EACH COMMON BLOCK REFERENCED IS UNIQUE FOR EACH BLOCK. THE BASES USUALLY START AT 2. FOR INSTANCE, IN THE EXAMPLE OF -SUBB-, THE SUBROUTINE (-SUBB-) REFERENCED FIFTEEN DIFFERENT COMMON BLOCKS. THE BASES FOR THESE BLOCKS WERE FROM 2 TO 20 (OCTAL), HENCE THE RELOCATION BASIS OF 21 (OCTAL) FOR EXTERNAL SYMBOLS. HAD -SUBB- ONLY REFERENCED TWO COMMON BLOCKS, THEIR BASES WOULD HAVE BEEN 2 AND 3, AND THE RELOCATION BASIS FOR EXTERNAL SYMBOLS WOULD HAVE BEEN 4.

EXAMPLE 6

```
MEMORY MAP:   NAME      LOCATION   LENGTH
              C           30000       332
SOURCE DECK STATEMENT:
              COMMON /C/IQ(218)
              DATA (IQ(1)=3,6)
     NOTE: 332 (OCTAL) = 218 (DECIMAL)
MEMORY DUMP:
     LOCATION 30000  00000000000000000003
              30001  00000000000000000006
```

108.4.4. MEMORY MAPS

MEMORY MAPS ARE PRINTED ON THE HSP OUTPUT AT THE END OF THE LOADING PHASE, WHETHER THE LOAD WAS SUCCESSFUL OR IN ERROR. IF IN ERROR, THE ADDRESSES OF BLANK AND/OR NUMBERED COMMON BLOCKS MAY NOT BE CORRECT FOR NON-CHAIN JOBS. THEY WILL BE INCORRECT IF THE ERROR WAS DETECTED ON THE FIRST PASS (I.E., THE FIRST NUMBERED COMMON BLOCK SEEN BY THE LOADER WILL HAVE A RELOCATION ADDRESS OF ZERO (O) INSTEAD OF AN ADDRESS HIGHER THAN THE LAST SUBROUTINE LOADED).

IF A CHAIN JOB IS NOT SUCCESSFULLY LOADED, THE MEMORY MAP ADDRESSES OF THE LINK IN ERROR WILL BE INCORRECT FOR THE SUBROUTINES AND LABELED COMMON BLOCKS IF THE ERROR WAS DETECTED ON THE FIRST PASS. IN THIS CASE THE LOADER HAS NOT YET DECIDED THE TOTAL CODE SIZE AND THE ADDRESSES WILL BE HIGHER THAN THAT OF A SUCCESSFUL LOAD, I.E., THE LOADER ASSUMES MAXIMUM MEMORY SIZE (272700 OCTAL) DURING THE FIRST PASS.

108.4.5. Monitor 402 Locations

The following locations within the file - MON400 - may be of interest. The locations are relative to work Zero (0), i.e.-,after the 400 (octal) minus words of the -MON400- file, and are stated in octal. These same locations and their functions also apply to monitor generated controllees.

LOCATION (OCTAL)

0	USED TO ISSUE FROST CALLS
	USED TO ISSUE I/O REQUESTS
2	CONTROL SENT HERE ON INDEFINITES OR INFINITE HARDWARE INTERRUPTS.
3-14	NOT USED.................
15	FORMAT OF DUMP IF ONE IS TO BE TAKEN
16-17	NOT USED.................
20-21	USED BY LRLLIB SUBROUTINE -STOPSW- INCREMENT LEFT & SSWI FLAG
22	BACKGROUND OPERATION MODE FLAG
23	TIME LEFT (USED BY -STOPSW-)
24	UPPER 30 BITS, POINTER TO CURRENT POSITION ON LOGICAL UNIT 14 (i.e., PUNCH CARD FILE)LOWER 30 BITS, POINTER TO POSITION ON INPUT FILE (LOGICAL UNIT 2).
25	LAST WORD ADDR$SS OF RESIDENT MONITOR ROUTINES (USED BY ONMON-OFFMON)
26	NOT USED
27	SENSE SWITCH, SENSE LIGHTS STATUS. BITS 0-5 ARE SENSE LIGHTS 1-6: BIT 59 USED BY -STOPSW-
30	CONTROLLEE(THIS) NAME IN ASCII BOD
31	NOT USED.......
32	FIRST WORD ADDRESS + 1 OF EITHER EXITH OR QUITH (LRLLIB SUBROUTINES
33	ERROR NUMBER FOR S.E.P. (STANDARD ERROR PROCEDURE)
34	NOT USED
35	FIRST WORD ADDRESS OF COMMON BLOCK -MINCOM-
36	DATE IN ASCII BCD G07/22 IS FORMAT (RIGHT ADJUSTED)
37	TIME IN ASCII BCD 13.021 IS FORMAT (RIGHT ADJUSTED)

NOTE: THESE TWO (36&37) WILL BE CURRENT AFTER A CALL TO THE LRLLIB SUBROUTINE -TIME-

40..............................LOADER/EXECUTIVE COMMUNICATION WORD
41..............................THIS CONTROLLEE'S FIELD LENGTH
42-45..............................HSP OUTPUT ID
46..............................STATISTICS USE......
47..............................INPUT FILE RECORD (OR CARD) COUNTER
50..............................FLAG SET BY SUBROUTINE -CHAIN- FOR S.E.P. WHEN ERROR READING CHAIN FILE.

-LRLLIB- IS THE NAME OF THE MONITOR LIBRARY FILE.

108.5. LOAD TYPES

108.5.1. INTRODUCTION

THE 402 LOADER DISTINGUISHES THE FOLLOWING NINE LOAD TYPES:

1. CONVENTIONAL (I.E., NON-CHAIN, NORMAL EXECUTION)
2. ZEROLOAD (NOT RECOMMENDED)
3. CONTROLLEE
4. ZEROLOAD-CONTROLLEE
5. CONVENTIONAL CHAIN
6. CONVENTIONAL M-O-S (I.E., MAIN-OVERLAY-SEGMENT CHAIN TYPE)
7. CONTROLLEE CHAIN
8. CONTROLLEE M-O-S
9. ZEROLOAD-CONTROLLEE M-O-S (NOT RECOMMENDED WITHOUT MORE DETAILED KNOWLEDGE)

(1) NON-CHAIN LOADS:

ALL LOADS EXCEPT CHAIN LOADS OF TYPES 5-8 WILL HAVE THE FOLLOWING MEMORY ALLOCATION:

(A) INSTRUCTION AND LABELED COMMON BLOCKS ARE INTERMIXED AT THE LOW ORDER ADDRESSES. THEY WILL BE LOADED IN ORDER OF APPEARANCE, WITH SUCCESSIVE FIRST WORD ADDRESSES IN ASCENDING ORDER.

(B) BLANK AND NUMBERED COMMON WILL START IMMEDIATELY AFTER THE LAST INSTRUCTION OR LABELED COMMON BLOCK. THEY WILL ALSO BE LOADED IN ORDER OF APPEARANCE, WITH SUCCESSIVE FIRST WORD ADDRESSES IN ASCENDING ORDER.

EXAMPLE 1:

SUPPOSE A JOB DEFINES CODE AND COMMON BLOCKS IN THE FOLLOWING ORDER:

```
SUBROUTINE A   - LENGTH P1
COMMON BLOCK 2 - LENGTH C1
BLANK COMMON   - LENGTH C2
SUBROUTINE B   - LENGTH P2
COMMON BLOCK C - LENGTH P3
COMMON BLOCK 1 - LENGTH C3
SUBROUTINE D   - LENGTH P4
```

C = C1+C2+C3, P = P1+P2+P3+P4.

108.5. LOAD TYPES

THEN THE LOAD WILL BE IN THIS ORDER:

```
FIRST WORD ADDRESSES:     0 ***************************
                              RESIDENT SYSTEM ROUTINES
                          N ***************************
                              SUBROUTINE A
                       N+P1 ***************************
                              SUBROUTINE B
                    N+P1+P2 ***************************
                              COMMON BLOCK C
                 N+P1+P2+P3 ***************************
                              SUBROUTINE D
                        N+P ***************************
                              COMMON BLOCK 2
                     N+P+C1 ***************************
                              BLANK COMMON
                  N+P+C1+C2 ***************************
                              COMMON BLOCK 1
                      N+P+C ***************************
```

NOTE: THIS ALLOWS ONE TO GIVE UP MEMORY FROM THE HIGH ORDER END OF THE FILE AND, CONVERSELY, TO INCREASE THE FILE LENGTH.

(2) ALL CHAIN LOADS EXCEPT TYPE 9:

LOADS OF TYPES 5-8 HAVE THE FOLLOWING MEMORY ALLOCATION:

(A) THE RESIDENT I/O AND SYSTEM ROUTINES WILL BE AT THE LOW ORDER ADDRESSES.

(B) BLANK AND NUMBERED COMMON ARE LOADED NEXT. THEY ARE LOADED IN ORDER OF APPEARANCE, WITH SUCCESSIVE FIRST WORD ADDRESSES IN ASCENDING ORDER.

(C) ASSUMING INSTRUCTION AND LABELED COMMON BLOCKS OF TOTAL SIZE P, THESE WILL BE LOADED FROM ADDRESS M TO M+P, WHERE M IS THE FIRST ADDRESS NOT ASSIGNED TO BLANK OR NUMBERED COMMON. HOWEVER, THEY ARE LOADED WITH FIRST WORD ADDRESSES IN DESCENDING ORDER. (THE ADDRESSES WITHIN AN INSTRUCTION OR LABELED COMMON BLOCK, OF COURSE, ARE STILL IN INCREASING ORDER.)

108.5. LOAD TYPES

EXAMPLE 2:

ASSUMING THE SAME ROUTINES AND COMMON BLOCKS AS ABOVE, THE TYPE 5-8 LOAD WILL BE IN THIS ORDER:

```
FIRST WORD ADDRESSES:    0 ***************************
                              RESIDENT SYSTEM ROUTINES
                         N ***************************
                              COMMON BLOCK 2
                      N+C1 ***************************
                              BLANK COMMON
                   N+C1+C2 ***************************
                              COMMON BLOCK 1
                       N+C ***************************
                              SUBROUTINE D
                    N+C+P4 ***************************
                              COMMON BLOCK C
                 N+C+P4+P3 ***************************
                              SUBROUTINE B
              N+C+P4+P3+P2 ***************************
                              SUBROUTINE A
                     N+C+P ***************************
```

NOTE: IN THIS CASE, THE USER CANNOT GIVE UP MEMORY UNLESS THE INSTRUCTIONS THERE ARE NEVER TO BE USED AGAIN.

IF A CONTROLLEE LOAD IS SPECIFIED, AN EXECUTABLE FILE, WITH THE NAME SPECIFIED ON THE /* CONTROLLEE/ CARD, WILL BE GENERATED BY MONITOR. THE EXECUTION OF THIS FILE IS TO BE INITIATED BY THE USER FROM HIS TELETYPE; IT WILL NOT BE EXECUTED BY MONITOR.

IF A ZEROLOAD (/* ZEROLOAD/ CARD) IS SPECIFIED, THE NEEDED RESIDENT I/O AND SYSTEM ROUTINES WILL BE LOADED FROM THE MONITOR LIBRARY. THIS MAY RESULT IN A SHORTER FILE THAN A NON-ZEROLOAD, BUT NOT USUALLY.

EXAMPLE 3:

IN THE FOLLOWING, THE SAME CODE IS SHOWN, LOADED AS TYPE 1, 2, 3, AND 4. ASSUME USER CODE, LIBRARY ROUTINES, AND LABELED COMMON BLOCKS OF LENGTH M; RESIDENT I/O AND SYSTEM ROUTINES OF LENGTH N; BLANK AND NUMBERED COMMON OF LENGTH P. L IS A NUMBER < M+N, AND T IS THE TOTAL LENGTH OF THE FILE.

108.5. LOAD TYPES

108.5.2. CONVENTIONAL

TYPE 1: NON-ZEROLOAD, NORMAL EXECUTION.

```
FIRST WORD ADDRESSES:     0 ***************************
                              RESIDENT SYSTEM ROUTINES
                          N ***************************
                              USER CODE AND LABELED
                              COMMON BLOCKS
                        M+N ***************************
                              BLANK AND NUMBERED
                              COMMON
                      M+N+P ***************************
                                       T=M+N+P
```

108.5.3. ZEROLOAD

TYPE 2: ZEROLOAD, NORMAL EXECUTION.

```
FIRST WORD ADDRESSES:     0 ***************************
                              RESIDENT SYSTEM ROUTINES
                          N ***************************
                              USER CODE AND LABELED
                              COMMON BLOCKS AND NEEDED
                              SYSTEM ROUTINES
                        L+N ***************************
                              BLANK AND NUMBERED
                              COMMON
                      L+N+P ***************************
                            M+N+P < T = L+N+P < M+2N+P
```

NOTE: THE USER IS NOT LINKED TO THE RESIDENT ROUTINES, SO THE FIRST N WORDS IS WASTED MEMORY.

108.5.4. CONTROLLEE

TYPE 3: NON-ZEROLOAD, CONTROLLEE.

SAME AS TYPE 1, EXCEPT FILE IS AN INDEPENDENT CONTROLLEE.

108.5.5. ZEROLOAD-CONTROLLEE

TYPE 4: ZEROLOAD-CONTROLLEE.

```
FIRST WORD ADDRESSES:    0 ****************************
                             USER CODE AND LABELED
                             COMMON BLOCKS AND
                             NEEDED SYSTEM ROUTINES
                         L ****************************
                             BLANK AND NUMBERED
                             COMMON
                       L+P ****************************
                                 T=L+P < M+N+P
```

108.5. LOAD TYPES

108.5.6. CONVENTIONAL CHAIN

TYPE 5: CONVENTIONAL CHAIN

IN THE CASE OF A CONVENTIONAL CHAIN LOAD (TYPE 5), ALL LINKS ARE FORCED TO THE SAME LENGTH AS THE LONGEST ONE. THE FIRST WORD ADDRESS OF THE BLANK/NUMBERED COMMON AREA IS THE SAME FOR ALL LINKS. ONLY THE INSTRUCTION/LABELED COMMON AREA IS WRITTEN ON THE CHAIN TAPE (OR FILE).

EXAMPLE 4:

```
                LINK 1                    LINK 2                    LINK 3

         0 *****************       0 *****************       0 *****************
              RESIDENT                  RESIDENT                  RESIDENT
         N *****************       N *****************       N *****************
              BLANK/NUMBERED            BLANK/NUMBERED
              COMMON                    COMMON                    BLANK/NUMBERED
      N+P1 *****************    N+P2 *****************            COMMON
                (GAP)                     (GAP)
   N+P1+G1 *****************  N+P2+G2 *****************    N+P3 *****************
              INSTRUCTION               INSTRUCTION               INSTRUCTION
         T *****************       T *****************       T *****************
                      T = N+P1+G1+M1 = N+P2+G2+M2 = N+P3+M3
```

108.5.7. CONTROLLEE CHAIN

TYPE 7: CONTROLLEE CHAIN

WITH THE SAME CODE, A CONTROLLEE CHAIN (TYPE 7) WOULD BE THE SAME AS THE ABOVE EXCEPT THAT ONE OF THE LINKS IS DESIGNATED AS A CONTROLLEE. THE CONTROLLEE FILE IS THE FULL LENGTH OF THE LONGEST LINK.

108.5.8. CONVENTIONAL M-O-S

TYPE 6: CONVENTIONAL M-O-S

IN THE CASE OF A CONVENTIONAL MAIN-OVERLAY-SEGMENT LOAD (TYPE 6), ALL LOADS ARE FORCED TO THE SAME LENGTH AS THE LONGEST EXECUTABLE LOAD. AS WITH TYPE 5, THE FIRST WORD ADDRESS OF THE BLANK/NUMBERED COMMON AREA IS THE SAME FOR ALL LOADS. ONLY THE INSTRUCTION/LABELED COMMON BLOCKS FOR EACH LINK ARE WRITTEN ON THE CHAIN TYPE (OR FILE).

EXAMPLE 5:

IN THIS AND THE FOLLOWING EXAMPLES, ASSUME THE FOLLOWING JOB INPUT:

LINK	TYPE	INSTRUCTION LENGTH	BLANK/NUMBERED COMMON
1	MAIN	M1	P
2	OVERLAY	M2	P
3	OVERLAY	M3	P1 (P1 > P)
4	SEGMENT	M4	P
5	SEGMENT	M5	P
6	OVERLAY	M6	P
7	SEGMENT	M7	P

EACH OF THE FOLLOWING IS A MAXIMUM EXECUTABLE LOAD.

```
             LOAD 1                                  LOAD 2

       0 *****************                0 *****************
           RESIDENT                           RESIDENT
       N *****************                N *****************
           BLANK/NUMBERED                     BLANK/NUMBERED
           COMMON                             COMMON
     N+P *****************             N+P1 *****************
               (GAP)                              (GAP)
                                        N+P1+G2 *****************
  N+P+G1 *****************                    LINK 4 SEGMENT
           LINK 2 OVERLAY               N+P1+S2 *****************
                                              LINK 3 OVERLAY
  N+P+O1 *****************              N+P1+O2 *****************
           LINK 1 MAIN                        LINK 1 MAIN
       T *****************                T *****************
```

108.5. LOAD TYPES

```
              LOAD 3                                LOAD 4

          0 *****************               0 *****************
              RESIDENT                          RESIDENT
          N *****************               N *****************
              BLANK/NUMBERED                    BLANK/NUMBERED
              COMMON                            COMMON
        N+P *****************             N+P *****************
                                                  (GAP)
              LINK 5 SEGMENT           N+P+G4 *****************
                                                LINK 7 SEGMENT
     N+P+M5 *****************          N+P+S4 *****************
              LINK 3 OVERLAY                    LINK 6 OVERLAY
     N+P+O3 *****************          N+P+O4 *****************
              LINK 1 MAIN                       LINK 1 MAIN
          T *****************               T *****************
```

HERE O1 = G1+M2, S2 = G2+M4, O2 = S2+M3, O3 = M5+M3, S4 = G4+M7,
O4 = S4+M6, AND T = N+P+M5+M3+M1 (THE SAME FOR ALL LOADS).

108.5.9. CONTROLLEE M-O-S

TYPE 8: CONTROLLEE M-O-S

IF THE MAIN LINK (LINK 1 IN EXAMPLE 5) OF A M-O-S WERE DESIGNATED AS A CONTROLLEE (LOAD TYPE 8), THE LOADS WOULD BE AS ABOVE, AND THE CONTROLLEE FILE WOULD BE FULL LENGTH OF THE LONGEST LOAD (LOAD 3 IN THE EXAMPLE). ONLY THE RESIDENT SYSTEM ROUTINES AND THE MAIN LINK WOULD BE IN THE FILE, THE REMAINDER BEING GARBAGE.

EXAMPLE 6:
THE CONTROLLEE FILE WOULD LOOK LIKE:

```
    0 *************
        RESIDENT
    N *************
        (GARBAGE)
 T-M1 *************
        LINK 1
        MAIN
    T *************
```

108.5.10. ZEROLOAD-CONTROLLEE M-O-S

TYPE 9: ZEROLOAD-CONTROLLEE M-O-S

IN CASE OF A LOAD OF TYPE 9 (ZEROLOAD-CONTROLLEE M-O-S), THE NEEDED RESIDENT I/O AND SYSTEM ROUTINES ARE LOADED FROM THE MONITOR LIBRARY WITH THE MAIN LINK. THE FIRST WORD ADDRESS OF THE BLANK/NUMBERED COMMON AREA WILL BE THE SAME FOR ALL LOADS. IF THIS LENGTH IS THE SAME LENGTH FOR ALL LINKS, ALL LOADS WILL BE THE SAME LENGTH AS THE LONGEST ONE. THE MAIN LINK + NEEDED RESIDENT ROUTINES WILL BE WRITTEN ON THE CHAIN TAPE (OR FILE) AS A LINK, AS WILL THE INSTRUCTION/LABELED COMMON BLOCKS FOR THE OVERLAY AND SEGMENT LINKS. THE CONTROLLEE FILE WILL HAVE LENGTH ONLY EQUAL TO THAT OF THE MAIN LINK + NEEDED RESIDENT ROUTINES.

108.5. LOAD TYPES

EXAMPLE 7:

LOAD 1

```
    0 *****************
         MAIN LINK +
         RESIDENT
    L *****************
         LINK 2 OVERLAY

 L+M2 *****************
            (GAP)

 TO-P *****************
         BLANK/NUMBERED
         COMMON
   TO *****************
```

LOAD 2

```
       0 *****************
            MAIN LINK +
            RESIDENT
       L *****************
            LINK 3 OVERLAY
    L+M3 *****************
            LINK 4 SEGMENT
 L+M3+M4 *****************
               (GAP)
   T1-P1 *****************
            BLANK/NUMBERED
            COMMON

      T1 *****************
```

LOAD 3

```
    0 *****************
         MAIN LINK +
         RESIDENT
    L *****************
         LINK 3 OVERLAY
 L+M3 *****************

         LINK 5 SEGMENT


 TO-P *****************
         BLANK/NUMBERED
         COMMON
   TO *****************
```

LOAD 4

```
       0 *****************
            MAIN LINK +
            RESIDENT
       L *****************
            LINK 6 OVERLAY

    L+M6 *****************
            LINK 7 SEGMENT
 L+M6+M7 *****************
               (GAP)
    TO-P *****************
            BLANK/NUMBERED
            COMMON
      TO *****************
```

IN THE ABOVE, L < M1+N, TO = L+M3+M5+P < T, AND TO-P = T1-P1. THE CONTROLLEE FILE WILL HAVE LENGTH L.

NOTE: IF BLANK/NUMBERED COMMON IS ABSENT OR IF ITS LENGTH VARIES FROM LOAD TO LOAD, THE EXECUTABLE LOAD LENGTH WILL VARY ACCORDINGLY.

108.6. UPDATING CHAIN TAPES

THERE IS AVAILABLE A LIMITED ABILITY TO UPDATE CHAIN FILES ON THE 6600. A LINK WILL BE UPDATED SUCCESSFULLY IF IT DOES NOT REFERENCE ENTRY POINTS AND/OR LABELED COMMON BLOCKS WITHIN ANY OTHER LINK. IF MORE THAN ONE LINK IS TO BE UPDATED THEY MUST BE CONSECUTIVE WITH RESPECT TO THE OLD CHAIN FILE. IF THE LINK TO BE UPDATED (I.E., RELOADED) IS NOT ON THE OLD FILE IT WILL BE ADDED TO THE END OF THE EXISTING FILE. IT IS UP TO THE USER TO SEE THAT THE NEW FILE IS LONG ENOUGH TO ACCOMMODATE THE ADDITIONAL LINK(S). FOR ALL PRACTICAL PURPOSES, THE CONVENTIONAL CHAIN LOAD TYPE IS THE ONLY TYPE OF CHAIN JOB THAT CAN BE SUCCESSFULLY UPDATED.

IT MAY BE THAT IN THE FUTURE, ANY TYPE OF CHAIN FILE CAN BE UPDATED. HOWEVER, THIS IS A MAJOR UNDERTAKING AND INVOLVES A MAJOR CHANGE IN THE FORMAT OF THE CHAIN FILE. WHEN TIME PERMITS IT WILL BE LOOKED INTO.

THE CONTROL CARD NECESSARY FOR AN UPDATE IS:

```
COLUMN    1     7
          *     UPDATE(N)
```

WHERE 'N' IS THE LOGICAL UNIT NUMBER OF THE OLD FILE. THE NEW FILE WILL BE ON THE LOGICAL UNIT NUMBER SPECIFIED ON THE /* CHAIN(L,NN)/ CARD, WHERE 'NN' IS THE LOGICAL UNIT NUMBER.

EXAMPLE:

TO UPDATE LINK 2 ON LOGICAL UNIT 5:

```
*     ASSIGN(5,0,NEWFILE,XXXXXX)
*     ASSIGN(7,0,OLDFILE,0)
*     UPDATE(7)
*     CHAIN(2,5)
```

NOTE: THIS EXAMPLE IS FOR BOTH OLD AND NEW CHAIN FILES ON DISC.

108.7. THE LOADING PROCESS

THE LOADER PERFORMS TWO PASSES IN ITS EXECUTION. THE FIRST PASS CONCERNS ITSELF WITH DETERMINING THE TOTAL SIZE OF THE LOAD AND THE RELATIVE SIZES OF INSTRUCTION AND LABELED COMMON AREA VERSUS BLANK AND NUMBERED COMMON AREA. IF MISSING SUBROUTINES ARE DETECTED AND A /* DEBUG/ CONTROL CARD IS ABSENT, OR IF THE TOTAL SIZE IS TOO LARGE, THE LOADING PHASE IS TERMINATED AT THE END OF THE FIRST PASS. ANY LIBRARY FILES SPECIFIED HAVE ONLY THEIR DIRECTORIES READ ON THE FIRST PASS. FOR OTHER INPUT, ONLY THE CONTROL INFORMATION IS PROCESSED DURING THE FIRST PASS. ERROR CHECKING FOR CARD SEQUENCE IS PERFORMED DURING THE FIRST PASS AND ONLY THE FIRST PASS. THE DIVISION INTO TWO PASSES ALLOWS FOR DETECTING OBVIOUS ERRORS WHILE THE LOADER ITSELF CAN EXECUTE AT A REDUCED SIZE.

DURING THE SECOND PASS, THE ACTUAL LOADING, LINKAGE, AND RELOCATION IS PERFORMED. CHAIN FILES AND MAPS ARE ALSO WRITTEN DURING THIS PASS. MORE SUBTLE ERRORS AND INCONSISTENCIES ARE DETECTED DURING THIS PASS.

BECAUSE SOME ADDRESSES ARE UNKNOWN DURING THE FIRST PASS, THE MEMORY MAP PRINTED WHEN AN ERROR IS DETECTED DURING THE FIRST PASS MAY NOT ACCURATELY REFLECT THE ADDRESSES OF A SUCCESSFUL LOAD. FOR NON-CHAIN LOADS THE INCORRECT ADDRESSES WILL BE THOSE FOR BLANK AND NUMBERED COMMON BLOCKS. FOR CHAIN LOADS THE INSTRUCTION AND LABELED COMMON ADDRESSES MAY BE INCORRECT. *(16)*

THE EXPANSION OF THE MEMORY SIZE OF THE LOADER DURING THE SECOND PASS REFLECTS THE SIZE OF THE INSTRUCTION PLUS LABELED COMMON AREA OF OF THE PROGRAM BEING LOADED. PRIOR TO EXECUTION OF THE PROGRAM, AN EXPANSION OR REDUCTION OF MEMORY OCCURS DEPENDENT ON THE SIZE OF THE BLANK AND NUMBERED COMMON AREA RELATIVE TO THE SIZE OF THE LOADER ITSELF.

----

(16) SEE 108.4. ON PAGE 12 FOR MORE INFORMATION ON MEMORY MAPS.

108.8. BINARY CARD DESCRIPTION

THE READER IS REFERRED TO THE REPORT CIC-MP-14 (EDITION 2) FOR A COMPLETE PHYSICAL DESCRIPTION OF THE BINARY CARDS.

108.8.1. -IDC- CARD (SUBPROGRAM IDENTIFICATION CARD)

THIS IS THE FIRST CARD IN THE BINARY DECK. THERE IS ONLY ONE PER SUBROUTINE. THE PROGRAM NAME IN WORD THREE IS THE SAME NAME THAT APPEARS ON THE 'SUBROUTINE NAME' CARD IN THE SOURCE DECK. THIS CARD ALSO DEFINES THE BYTE SIZES OF THE CONTROL INFORMATION APPEARING ON THE REMAINDER OF THE BINARY CARDS. THIS RESULTS IN AS SMALL A DECK AS POSSIBLE.

108.8.2. -SEG- CARD (SEGMENT IDENTIFICATION CARD)

THE SEGMENT CARD(S) FOLLOW THE -IDC- CARDS AND DEFINE SEPARATELY RELOCATABLE BLOCKS WITHIN THE SUBPROGRAM. THIS CARD IS NOT CURRENTLY USED. ITS FUTURE WILL PROBABLY BE IN CONNECTION WITH PL/I COMPILED CODES. A LENGTH IS DEFINED ON THIS CARD AND THE LENGTH DEFINED ON THE PRECEDING -IDC- CARD TAKES THIS INTO CONSIDERATION SO THAT THE LENGTH IS NOT COUNTED TWICE. THE BLOCKS DEFINED ARE UN-NAMED BUT DO DEFINE RELOCATION BASES DIFFERENT FROM THE ONE DEFINED ON THE -IDC- CARD. THESE BLOCKS ARE RELOCATED IN MEMORY APART FROM THE SUBPROGRAM WHICH DEFINES THEM.

108.8.3. -EPT- CARDS (ENTRY POINT CARD)

THESE CARDS DEFINE UNIQUELY NAMED TRANSFER ADDRESSES WITHIN A SUBPROGRAM SO THAT OTHER SUBPROGRAMS MAY DIRECTLY CALL THEM. THESE NAMES WOULD APPEAR AS EXTERNAL SYMBOLS IN OTHER SUBPROGRAMS. THESE ENTRY POINT NAMES DO NOT CONSTITUTE SEPARATELY RELOCATABLE BLOCKS. THERE IS NO LIMIT AS TO THE NUMBER OF THESE CARDS, HOWEVER THE FIRST BLANK FIELD ON A GIVEN CARD TERMINATES THAT CARD.

108.8. BINARY CARD DESCRIPTION

108.8.4. -BCT- CARD (BLOCK COMMON TABLE CARD)

THESE CARD(S) DEFINE COMMON BLOCK NAME(S), LENGTH(S), AND THE RELOCATION BASES ASSOCIATED WITH EACH NAME. THE CURRENT UPPER LIMIT IS 127(10) UNIQUE NAMES.

108.8.5. -PRM- CARD (PARAMETER VALUE CARD)

THIS CARD DEFINES THE VALUE OF A GIVEN PARAMETER NAME AND DEFINES THE RELOCATION BASIS ASSOCIATED WITH IT. IT IS NOT TO BE CONFUSED WITH THE PARAMETER STATEMENT IN THE /LRLTRAN/ COMPILER. IT PROVIDES A MEANS OF DEFINING AN EXTERNAL SYMBOL OTHER THAN COMMON NAMES AND ENTRY POINTS. THIS CARD IS NOT CURRENTLY IMPLEMENTED BY THE /LRLTRAN/ COMPILER.

108.8.6. -EXT- CARD (EXTERNAL SYMBOL CARD)

THIS CARD CONTAINS THE BCD NAMES OF ALL SYMBOLS EXTERNAL TO (I.E., NOT DEFINED WITHIN) THE SUBPROGRAM. THE RELOCATION BASIS ON THE -RBD- CARDS IS THE SAME FOR ALL SUCH EXTERNAL SYMBOLS. THE ORDER OF THE NAMES ON THIS CARD TOGETHER WITH THE -RAX- AND -LIT- CARDS DEFINES THE ORDER OF THE TABLE OF EXTERNAL SYMBOL VALUES COMPUTED BY THE LOADER. THE ADDRESSES ON THE -RBD- CARDS, OF THE WORDS WHOSE RELOCATION BASIS IS AN EXTERNAL SYMBOL BASIS, POINT TO ENTRIES WITHIN THIS LOADER COMPUTED TABLE.

108.8.7. -LIT- CARD (LITERAL VALUE CARD)

THIS CARD DEFINES A LITERAL VALUE, WHICH CAN BE RELOCATED BY SOME ALREADY DEFINED BASIS, TO BE USED EITHER IN THE COMPUTATION OF AN EXPRESSION USING EXTERNAL SYMBOLS (-RAX- CARD) OR TO BE DIRECTLY REFERENCED BY THE SUBPROGRAM. THIS CARD DEFINES THE POSITION OF THE VALUE IN AN INTERNAL TABLE AND THIS POSITION IS POINTED TO EITHER BY THE -RAX- EXPRESSION OR BY THE ADDRESS PORTION OF A RELOCATABLE WORD ON AN -RBD- CARD.

Errata 1
October 15, 1968

108.8.8. -RAX- CARD (RELOCATABLE ADDRESS EXPRESSION CARD)

THIS CARD CONSISTS OF A STRING OF SYLLABLES (THE BYTE SIZE OF THE SYLLABLE IS DEFINED ON THE -IDC- CARD) WHICH IS A POLISH REPRESENTATION OF AN ARITHMETIC EXPRESSION INVOLVING THE VALUES OF EXTERNAL SYMBOLS AND TO BE COMPUTED BY THE LOADER. THE COMPUTED VALUE OF THESE EXPRESSIONS ARE STORED IN AN INTERNAL TABLE (THE POSITION IN THE TABLE IS DETERMINED BY A POINTER IN THE -RAX- EXPRESSION) WHICH IS POINTED TO BY THE ADDRESS PORTION OF THE RELOCATABLE WORD ON THE -RBD- CARD.

THIS CARD CAN ALSO CONTAIN SYLLABLE STRINGS DENOTING ARBITRARY BIT FIELD RELOCATION FOR ANY WORD ON THE RBD CARDS. IN THIS CASE OF BIT FIELD RELOCATION THE LOADER CONSTRUCTS AN ENTIRE WORD AND STORES IT INTO THE INTERNAL TABLE MENTIONED ABOVE. THE ENTRY IN THE INTERNAL TABLE CONTAINS A FLAG DENOTING WHOLE WORD REPLACEMENT. THIS FLAG IS CHECKED WHEN THIS ENTRY IS POINTED TO BY THE ADDRESS PORTION OF THE WORD TO BE REPLACED (WHICH IS ON AN -RBD- CARD).

THE CURRENTLY IMPLEMENTED OPERATORS AVAILABLE IN THE ARITHMETIC EXPRESSION REFFERRED TO IN THE FIRST PARAGRAPH OF THIS SECTION ARE UNARY MINUS, BINARY MINUS, PLUS, MULTIPLY, AND DIVIDE. THE OPERATORS FOR BIT FIELD RELOCATION ARE WIDTH (I.E. NUMBER OF BITS IN THE BYTE TO BE RELOCATED), OFFSET (RIGHTMOST BIT POSITION, RELATIVE TO THE ENTIRE WORD), AND A SEPARATOR SENTINAL USED TO DISTINGUISH BETWEEN DIFFERENT BYTES IN THE SAME COMPUTER WORD. THE WIDTH IS SET TO THE ADDRESS LENGTH OF THE COMPUTER WORD AS AN INITIAL DEFAULT CONDITION (ON THE 6600, EIGHTEEN BITS) AND THE OFFSET IS SET TO ZERO AS AN INITIAL DEFAULT CONDITION. IF EITHER THE WIDTH OR THE OFFSET OPERATORS ARE DEFINED IN ANY GIVEN EXPRESSION, THE SUBSEQUENT DEFAULT VALUE(S) WILL BE THAT OF THE LAST EXPLICIT DEFINITION.

OPERATORS (OCTAL REPRESENTATION)	FUNCTION
00	ILLEGAL
01	+
02	- (BINARY)
03	*
04	/
05	ADDRESS REPLACEMENT
06	- (UNARY)
07	BIT FIELD OFFSET
10	BIT FIELD WIDTH
11	BIT FIELD OPERATOR SENTINEL
12	FULL WORD REPLACEMENT
13	Pass Operator

108.8. BINARY CARD DESCRIPTION

108.8.9. -RBD- CARDS (RELOCATABLE BINARY DECK)

THESE CARDS CONTAIN THE OBJECT CODE OF THE PROGRAM (I.E., THE INSTRUCTIONS, LITERALS, AND VALUES DEFINED IN A FORTRAN 'DATA' STATEMENT). THE FIRST COMPUTER WORD ON THE CARD (ON THE 6600, SIXTY BITS OR FIVE COLUMNS) CONTAINS THE NUMBER OF RELOCATABLE WORDS ON THE CARD AND THE RELATIVE OR UNRELOCATED ADDRESS OF THE FIRST RELOCATABLE WORD. THE NEXT N COMPUTER WORDS CONTAIN RELOCATION INFORMATION FOR EACH RELOCATABLE WORD ON THE CARD (ON THE 6600, THERE IS RELOCATION INFORMATION FOR EACH HALF WORD). THE VALUE OF N IS DETERMINED BY THE SIZE OF THE BYTE CONTAINING THE RELOCATION INFORMATION AND THE NUMBER OF RELOCATABLE WORDS ON THE CARD. SEE THE PUBLICATION CIC-MP-14 (EDITION 2) FOR MORE DETAIL. SINCE EACH CARD CONTAINS ITS OWN FIRST WORD ADDRESS, THESE CARDS (-RBD- CARDS ONLY) CAN BE OUT OF ORDER.

108.8.10. -TRA- CARD (END CARD)

THIS CARD MERELY SERVES AS AN END SENTINAL FOR THE RELOCATABLE SUBROUTINE.

108.9. LOADER ERROR AND WARNING COMMENTS

IN THE FOLLOWING ERROR COMMENTS, 'F' MEANS FATAL ERROR (LOADING CANNOT CONTINUE) AND 'NF' MEANS NON-FATAL (OR WARNING) TYPE OF ERROR COMMENT. THE FOLLOWING NOTATION IS USED:

-NAME-	IS THE BCD NAME OF AN ENTRY POINT OR PROGRAM.
-TAPE-	IS THE LOGICAL UNIT NUMBER OF THE TAPE OR FILE INVOLVED.
-TYPE-	IS THE CARD TYPE: -SEG-, -EPT-, -BCT-, -EXT-, -LIT-, -PRM-, OR -RAX-.
-X-	IS THE SEQUENCE NUMBER OF THE BINARY CARD, RELATIVE TO THE START OF THE SUBPROGRAM DECK. (FOR LIBRARY ROUTINES, NUMBERING STARTS WITH THE FIRST -RBD- CARD.)
-Y-	IS THE SEQUENCE NUMBER TAKEN FROM WORD 1 OF THE BINARY CARD.

REMARK: THE OCTOPUS DIAGNOSTIC 'PROBLEM ERROR 200' INDICATES A LOADER ERROR OR MACHINE ERROR.

THE FOLLOWING ERROR COMMENTS OCCUR ON-LINE, OFF-LINE (AFTER THE LOADER MAP) AND ON TTY:

CANNOT LOAD DATA INTO COMMON BLOCK -NAME- CARD -X- PROGRAM -NAME- (NF)

CANNOT OPEN OR CLOSE FILE -NAME- (F)
PROBABLE CAUSE: BAD TTY TRANSMISSION OR MIS-TYPE OF LIBRARY FILE NAME. (USUALLY REFERS TO TAPE 15.)

CANNOT READ FILE -TAPE- (F)
PROBABLE CAUSE: IF TAPE=15, BAD DISC FILE, BAD TTY TRANSMISSION, OR MIS-TYPE.

CANNOT WRITE FILE -TAPE- (F)

CARD SEQUENCE ERROR. PROGRAM -NAME- (F)
(A MORE DETAILED COMMENT APPEARS OFF-LINE, PRIOR TO MAP LISTING.)

END OF FILE ON CHAIN -TAPE- (F)
PROBABLE CAUSE: BAD TAPE OR TOO SHORT A DISC FILE.

EPT CARD MISSING IN PROGRAM -NAME- (F)

EXT OR LIT CARD MISSING IN PROGRAM -NAME- (F)

EXT, LIT, OR RAX CARD MISSING. PROGRAM -NAME- (F)

Errata 1
October 15, 1968

108.9. LOADER ERROR AND WARNING COMMENTS

FATAL CONFLICT WITH AN ENTRY POINT IN PROGRAM -NAME- (F)
PROBABLE CAUSE: (1) LOADER ERROR, OR (2) ATTEMPTING TO LOAD A SUBROUTINE WHICH HAS AN ENTRY POINT NAME IDENTICAL TO A RESIDENT SYSTEM ROUTINE OR TO A ROUTINE LOADED IN A PREVIOUS MAIN OR OVERLAY LINK.

IDC CARD MISSING. PRECEDING PROGRAM -NAME- (F)
PROBABLE CAUSE: (1) FIRST BINARY CARD FOLLOWING THE PROGRAM NAMED IS MISSING, OR (2) DECK HAS BEEN SHUFFLED.

ILLEGAL CARD TYPE. CARD -X- PROGRAM -NAME- (F)

INSTRUCTIONS + LABELED COMMON EXCEED LOADER LIMIT (F)
POSSIBLE REMEDY: MOVE SOME DATA FROM LABELED COMMON TO NUMBERED COMMON.

LITT POINTER OUT OF TABLE RANGE (F)
POSSIBLE CAUSE: (1) BAD BINARY DECK, (2) COMPILER OR ASSEMBLER ERROR, (3) PROGRAMMER ERROR, IF HAND-CODED, OR (4) TOO MANY LITERALS.

LOAD ADDRESS OUT OF BOUNDS. RBD CARD -X- PROGRAM -NAME- (F)

LOADER LIMIT REACHED. NUMBER OF EXT AND COMMON FROM LIBE TAPE. (F)

LOADER LIMIT REACHED. NUMBER OF SUBROUTINES FROM LIBE TAPE. (F)
NOTE: THE LOADER WILL LOAD A MAXIMUM OF ONE HUNDRED SUBROUTINES FROM EACH LIBE TAPE (I.E., 10 AND 15) FOR EACH LINK. THE CURRENT MAXIMUM FOR AN ENTIRE LIBE TAPE IS TWO HUNDRED SUBROUTINES.

MISSING SUBROUTINES (F)
(A LIST APPEARS OFF-LINE, PRIOR TO MAP LISTING.)

MULT. DEFINED RAX EXPRESSION. CARD -X- PROGRAM -NAME- (F)

MULTIPLY DEFINED ENTRY POINT FROM LIBRARY TAPE. ENTRY -NAME- PROGRAM -NAME- (F)
PROBABLE CAUSE: THE USER'S PROGRAM HAS DEFINED ONE OR MORE, BUT NOT ALL, OF THE ENTRY POINTS WITHIN A LIBRARY ROUTINE.

OVERLAP OF BIT FIELDS. RAX CARD -X- PROGRAM -NAME- (NF)

POLISH TABLE POINTER TOO LARGE. CARD -X- PROGRAM -NAME- (F)

PROGRAM AND COMMON OVERLAP. IN CARD -X- PROGRAM -NAME- (F)

RELOCATION BASIS VALUE TOO BIG. CARD -X- PROGRAM -NAME- (F)
POSSIBLE CAUSE: FORTRAN 400 BINARY DECK.

OPERATOR OUT OF RANGE. CARD -X- PROGRAM -NAME-
Cause: New no-op (pass) operator out of range. RAX card no good (F)

108.9. LOADER ERROR AND WARNING COMMENTS

RELOCATION BYTE TOO LONG. POSSIBLY NOT A 6600 DECK. PROGRAM -NAME- (F)
PROBABLE CAUSE: IF -NAME- LOOKS LIKE GARBAGE, NOT A 6600 DECK.

TRA CARD MISSING. PROGRAM -NAME- (F)
PROBABLE CAUSE: (1) PARTIAL BINARY DECK, OR (2) DECK HAS BEEN SHUFFLED.

TRANSFER ENTRY POINT MISSING (F)
PROBABLE CAUSE: MAIN. OR NAME SPECIFIED ON /* CONTROLLEE/, /* CHAIN/, /* MAIN/, /* OVERLAY/, OR /* SEGMENT/ CARD HAS NOT BEEN LOADED.

UNDEFINED RELOCATION BASIS. CARD -X- PROGRAM -NAME- (F)
POSSIBLE CAUSE: (1) BAD BINARY CARD, (2) COMPILER OR ASSEMBLER ERROR, (3) PROGRAMMER ERROR, IF HAND-CODED, OR (4) FORTRAN 400 BINARY DECK.

WIDTH OR OFFSET IS NEGATIVE. CARD -X- PROGRAM -NAME- (F)

WIDTH PLUS OFFSET OVER ONE WORD. CARD -X- PROGRAM -NAME- (F)

WIDTH SPECIFICATION TOO SMALL. CARD -X- PROGRAM -NAME- (F)

THE FOLLOWING DIAGNOSTICS APPEAR OFF-LINE ONLY, BEFORE THE LOADER MAP:

COMMON BLOCK -NAME- IN SUB. -NAME- IS OF INCONSISTENT LENGTH (NF)
MEANING: A SHORTER BLOCK OF THIS NAME HAS ALREADY BEEN ASSIGNED.

MULT. DEFINED ENTRY POINT -NAME- (NF)

-NAME- UNDEFINED (NF)
REMARK: IF /* DEBUG/ CARD IS ABSENT, THIS IS A FATAL ERROR, AND THE MESSAGE WILL APPEAR ON-LINE AS WELL.

-TYPE- CARD -Y- MISSING. LAST CARD READ HAS -X- RELATIVE TO START OF SUBROUTINE DECK. (F)

DISTRIBUTION

RE-RUN 7-7-72

TID File 200

NOTICE

"This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights."