

technical memorandum

Daresbury Laboratory

DL/CSE/TM03

A USERS GUIDE TO THE DARESBURY INTERDATA CROSS-LINKER

by

W.H. PURVIS, Daresbury Laboratory

FEBRUARY, 1977

Science Research Council

Daresbury Laboratory

Daresbury, Warrington WA4 4AD

LENDING COPY

by
W. H. PURVIS
Daresbury Laboratory

DARESBURY LABORATORY
SCIENCE RESEARCH COUNCIL

1. INTRODUCTION

The Daresbury Laboratory Interdata cross-linker allows Interdata users to generate a complete load module on the 370 from object modules produced by the CAL cross assembler. Facilities include automatic library call processing and production of module maps. Input is in Daresbury Laboratory Interdata object code (see Appendix A). Output is also in Daresbury Laboratory Interdata object code (loader subset).

2. JCL

The linker needs the following JCL for a batch run:

```
//UT EXEC PGM=LINK85,REGION=LOOK,TIME=(,59),  
// PARM='UPSI=OOL00000,options'  
//SYSPRINT DD SYSOUT=A  
//SYSLIB DD DSN=library,DISP=SHR (if library is used)  
//SYSLIN DD DSN=object,DISP=(OLD,DELETE)  
//SYSUT1 DD UNIT=SYSDA,SPACE=(TRK,(10,10))  
//SYSPUNCH DD SYSOUT=(P,PTAPE) (paper tape output)  
//SYSIN DD *  
linker control cards  
/*
```

SYSLIN is usually the output from a previous CAL assembly step. If not required it must be replaced by a DD DUMMY.

SYSIN defines the control card input. In order to process these cards it is necessary to include 'OPT' in the list of options in the parm field. If this is omitted, a simple linking procedure is followed and SYSIN may be omitted. Output is via SYSPUNCH and consists of 72 byte object records. These may be punched on paper tape, cards or mag. tape as desired. Output may be blocked if your loader can process blocked input.

3. CONTROL CARDS

If the 'OPT' keyword is included in the parm field, the linker will

read records from SYSIN and interpret them as linker control cards. Each card is assumed to contain a single command. Command format is simply:

Commandname Operands

where commandname may be one of the commands described below and the operands depend on the command. The following commands are currently available (a list is given in Appendix B).

INCLUDE item,.....

This command requests the linker read the object file defined by 'item'. This may be a sequential data set, in which case 'item' is the DDNAME provided to access that data set, or a member of a partitioned data set, in which case either the member name is specified on the DD card and 'item' is as for sequential data sets, or 'item' is of the form 'ddname(member)' and a DD card has been included for the data set. In this case the member name need not appear on the DD card.

ENTRY name

This command informs the linker that the user wishes the module to be entered at 'name'. 'name' must be an entry point into a module that has already been processed at the time the ENTRY command is read, otherwise an error message will be output and the command ignored. Note that this restricts entry points to defined entry points. Other entry points may be defined by means of the END statement in the CAL source, the linker taking the first of these as the default entry point if no ENTRY statement is supplied.

ALIGN number

This command is needed for modules to be run on Interdata 7/16s. Fullword alignment may be needed in this case and can be enforced for selected modules by using the ALIGN command with an operand of '4' before the INCLUDE statements that read in those modules. To revert to half word alignment 'ALIGN 2' may be entered. Note that modules read in from SYSLIN can only be full word aligned if the keyword 'ALIGN=4' is included in the PARM field on the EXEC card.

MAP

The MAP command causes the linker to print out a module map, including a list of unresolved references if appropriate. If the linker is being

used interactively, this can then be followed by further INCLUDE commands to resolve these references and the MAP command re-entered to produce an updated map. A map of the final module may be produced by including the keyword 'MAP' in the PARM field.

LET

This command will cause the linker to produce an output module even though errors may have occurred. This may be applicable during program development where a part of the program which has not yet been written may be omitted to produce a test version. This will work as long as the references which are missing are not used during the test.

END

This command is provided to terminate processing when the linker is being used interactively. The linker will terminate without producing any output. Under normal conditions the command processing terminates at end of file and the linker will then produce a load module provided all external references have been resolved or the 'LET' command or keyword has been supplied.

OVERLAY name

This command allows the user to generate overlay segments. When this command is read, an overlay record is added to the object code and the 'name' supplied is included as part of this record (see appendix). If the linker is already processing an overlay segment, the address at which the next module is to be loaded will be reset to be the same as the previous segment.

ROOT

This command terminates OVERLAY processing. The address for the next module is set to be the end of the longest segment processed. This allows the user to create several overlay areas, each with a number of segments.

4. PARM OPTIONS

There are a number of options which may be entered into the PARM field of the EXEC card. Some of these have already been mentioned above.

The following describes each one in turn:

LIST

This keyword causes the linker to print out each command as it is processed. Useful for batch runs.

MAP

This keyword causes the linker to produce a load module map on the SYSPRINT file. This will be produced after commands have been processed and library call resolution has been performed. It is therefore a complete map of the module that is output.

OPT

This keyword requests that SYSIN be read for linker commands. If it is omitted, the linker will still attempt to open SYSIN but it will not read from it. If SYSIN is not supplied, the linker will continue normally provided the OPT keyword has not been specified.

LET

This keyword will cause the linker to produce a load module despite errors or missing external references.

NCAL

This keyword suppresses the automatic library call facility. If not specified, the linker will search SYSLIB for members whose names correspond to missing references. Those members which are found are read in and the process repeated until either no more members can be found or all references are satisfied. If NCAL is specified, this processing is bypassed and the SYSLIB DD card may be omitted. The library call resolution takes place after all the linker commands have been processed.

ALIGN=4

This command is provided for users of Interdata 7/16 (or equivalents) where full word alignment is required for some instructions. It will cause each module to be aligned on a full word. If omitted modules are aligned on half word boundaries.

ALIGN=2

Is included for completeness, it causes modules to be aligned on half word boundaries.

Note that the PARM field scanning will ignore unrecognised keywords. The linker does produce a list of those keywords it recognises at the head of the SYSPRINT listing.

5. ERROR MESSAGES

On detecting an error, the linker will produce an error message. The error messages are listed below. Any of these messages will produce a return code of 8 when the linker terminates. Unresolved references will produce a return code of 4 provided no other errors have occurred.

invalid object record	input is not DL object code
invalid external symbol type	ditto.
multiple entry definition	the same name appears in two modules
invalid ESD reference	object module corrupted
unable to open SYSUT1	missing DD card probably
unable to open SYSLIN	ditto.
unable to open SYSPUNCE	ditto.
unknown command	command not in list given above
invalid symbol name	must be letters and digits and '.'
entry point not defined	must be defined before command
dictionary full	more than 500 names
unable to open include file	missing DD card probably
unable to open SYSLIB	ditto.
invalid member name	must be letters and digits
too many program sections	more than 200 modules
system error	linker bug - inform DL

6. MISCELLANEOUS POINTS

SYSUT1 is used as a temporary work file during the input processing phase. It should be large enough to hold the complete module in expanded

form - about 40 per cent larger than total input. The defaults given in the JCL above should be adequate for most purposes.

SYSPRINT has a record length of 121 bytes with record format FBA. If desired a BLKSIZE may be specified which is a multiple of 121. If not specified a default of 3146 is written into the linker. Output BLKSIZE defaults to 72 since most loaders do not perform unblocking. If output is to be cards or paper tape, it may be feasible to block the output to some multiple of 72. The record format is FB to allow blocking when desired.

The overlay facility (OVERLAY and ROOT commands) was added in response to user requests but is not supported by the normal loader.

If no command input is supplied, the basic sequence of actions is:

- 1) read SYSLIN
- 2) resolve missing references from SYSLIB
- 3) produce a map (if MAP keyword specified)
- 4) output a load module (if no errors or LET keyword specified)

APPENDIX A

DARESBUURY LABORATORY OBJECT CODE FORMAT

Object code consists of a series of 72-byte records in the following format:

pos.	size	description
0	1	X'FO' start of record marker
1	1	length of record
2	1	Checksum. see below
3	1	Record type - see below
4	-	data portion - see below

Although records are all 72 bytes long, only the number specified in the record header are significant. The checksum is determined by adding the significant bytes together then subtracting from 256 (modulo 256). Thus the record may be checked by adding together all the bytes including the checksum. The total (modulo 256) should be zero.

There are 4 different types of records:

record type 1 - External Symbol Descriptors

offset	size	description
+0	8	symbol name (ISO characters)
+8	1	symbol type: <ol style="list-style-type: none"> 1 program name 2 entry point (absolute) 3 entry point (relocatable) 4 external reference
+9	2	depends on symbol type: <ol style="list-style-type: none"> 1 size of relocatable section of program 2 address of entry point 3 address of entry point 4 field absent (only 9 bytes for this entry)

The above format is repeated until the end of record.

APPENDIX B
SUMMARY OF LINKER COMMANDS

Record type 2 - Object code

offset size description

+0 2 address at which the following data is to be loaded
+2 1 relocation for load address
+3 2 two bytes of data
+5 1 relocation for data

last 3 bytes are repeated up to end of record.

Record type 3 - END record.

offset size description

+0 8 compiler/assembler name (ISO characters)
+8 2 entry point from end card (0 if not spec)
+10 1 relocation of entry point (0 if not spec)
+11 8 date of compilation/assembly (ISO characters)

Record type 4 - Overlay header

offset size description

+0 8 Segment name (ISO characters from OVERLAY command)
+8 2 Address of start of segment
+10 1 relocation of segment address (should always be 1)

The relocation byte is 0 for absolute values, 1 for the current section. Values greater than 1 refer to external symbols which are assigned sequence numbers starting at 2 up to a maximum of 255.

The object code format was designed to be such that an object module for a self-contained program should be acceptable to the loader without a need for linking. This is in fact the case since the loader will ignore all type 1 records, load all type 2 records and stop after reading a type 3 record. The linker does not output type 1 records and all relocation is reduced to a binary decision (absolute/relocatable).

The following is a list of the commands described above.

INCLUDE 'item',...
ENTRY name
ALIGN number
MAP
LET
END
OVERLAY name
ROOT

