ND SOFTWARE LIBRARY DISKETTE

Containing :
BRF-Linker for ND-100

Directory Name :
210721C01-XX-01S

User Name : **FLOPPY-USER**

No	File name	Type	T	Public	Friend	Own	Pages	Bytes
0	**BRF-LINKER-C01**	**PROG**	**I**	**R**	**R**	**RWACD**	**45**	155648

1 file using 45 pages. 148 pages reserved out of 148 pages.

ND SOFTWARE LIBRARY DISKETTE

Containing :
BRF-Linker for ND-100

Directory Name :
210721C01-XX-01D

User Name : **FLOPPY-USER**

No	File name	Type	T	Public	Friend	Own	Pages	Bytes
0	**BRF-LINKER-C01**	**PROG**	**I**	**R**	**R**	**RWACD**	**45**	**155648**

1 file using 45 pages. 610 pages reserved out of 610 pages.

BRF-Linker for ND-100
210721C01
Date: 880304 Report no: 1

Program: BRF-linker
Title: Corrections in revision C01.

- -

This errors are correct in C01 revision of the
BRF-LINKER for ND-100 :
---

1. The overlay system has been corrected.
   When loading a program using overlay mode, an error did occur. The error message "NO SUCH PAGE" occured when you ran the program.

2. The BRF-LINKER are now runing on non-CX CPU's.
   When using BRF-LINKER-C00 on non-CX CPU the error message "ILLEGAL INSTRUCTION" occured.

Product	Name BRF-Linker for ND-100	Reg. no. 210721C	Category STPR

Reason	. New product X Change/Addition	X Error Correction . Different Environment
Documen-tation	Title BRF-LINKER User Manual	Reg. no. 60.196.2 EN
Purpose	Load and link BRF-units to program files, and editing of BRF files.	
Prerequi-sites	Computer Type Floating format Op. system Version ND-10/100 SIN III VS >=H	
	Minimum mass storage resources for installation User User space Number of files SYSTEM 44 pages on 1 file	
	Minimum permanent mass storage resources User User space Number of files SYSTEM 44 pages on 1 file (for SIN. H)	
		Reg. no. Source 210722C

File Name	Type	Containing
BRF-LINKER-C<rev>	PROG	BRF-Linker

NOTE: <rev> is to be replaced by the current revision of the DIRECTORY or FILE. The revision level is found on the preceding "ND SOFTWARE LIBRARY DISKETTE" pages.

1 ERRORS CORRECTED

- LOAD file-name,,,,,
  Error corrected when including comma (,) after the file name in the LOAD command.
- The scratch file can be used as input file in the editor commands (APPEND-BRF 100,destination,,).
- The REPLACE-BRF command can be used on files containing BRF-units including COMMON.
- Error corrected concerning load of files prepared with the command PREPARE-BRF-LIBRARY-FILE.
- The load will be terminated if the error message 'Program/Data space exceeded' is printed.
- Errors corrected in mixing short and long S-groups (4-bytes and 6-bytes symbol groups).

2 MODIFICATIONS

2.1 CHANGED COMMANDS

- DEBUG-MODE <ON/OFF>

  This command has been given a new parameter: EXTENDED ie.

  Brl: DEBUG-MODE <ON/OFF/EXTENDED>

  If you get the error message from the loader which reads

  Brl mesage: Debug table full

  then the command:

  Brl: DEBUG-MODE EXTENDED

  can be included in the command list as a new command before loading the file giving this error message.

  Using this command, the version F01 of SYMBOLIC DEBUGGER is needed.

- SEGMENT-ENTRY <symbol>,(<symbol>....)

  In addition to this command, the <symbol> parameter can be one of the fixed symbols *DATA*,*PROG* or *REF*.

  SEGMENT-ENTRY *DATA*

  All defined data symbols will be exported and can be linked-to from other program files.

  SEGMENT-ENTRY *PROG*

  All defined program symbols will be exported and can be linked-to from other program files.

  SEGMENT-ENTRY *REF*

  All symbols in the segment entry command following the *REF* symbol will be generated with reference to the symbol.

  Example:

  Brl: SEGMENT-ENTRY *REF*,symbol1,symbol2

  is equal to

  Brl: SEGMENT-ENTRY symbol1,symbol2
  Brl: REFERENCE symbol1,,p
  Brl: REFERENCE symbol2,,p

- IGNORE-ENTRY <symbol>,(<symbol>...)

  This command has been extended. If the first <symbol> in the list is *SELECT*, then the command will be treated as a SELECT command and not as IGNORE. This means that all symbols in the list will be loaded from the next library file, and all other symbols will be ignored (prevented from loading).

  IGNORE-ENTRY *SELECT*,<symbol>,(<symbol>...)

2.2 NEW COMMANDS

- DATA-FILE <file-name>

  Multisegment mode.

  This command is used to put the data-part of a multisegment system into the main program file. The parameter <file-name> must be a loaded multisegment program file, and it must be the file including the main program entry (main program file).

  The data load address will be automatically set for the current program file, and the data loaded will be appended to the data on the data file.

  The command must be the first to follow the program-file command. The data-file is not to be included in the LINK-TO command list, it will be automatically included as the first entry.

  The data will be loaded into the current program file, and later copied to the data file. The size of the data area for the program file will be set to 0 (zero).

  No calls between segments will change the data area.

  The link information for the data will be on the current program file.

Example:

@BRF-LINKER-C<rev> program-file main/s0 load .... exit	Load the main program file, this file are to be used as data file.
@BRF-LINKER-C<rev> program-file sub1/s1 DATA-FILE main load ... exit	Load a new program file. Data part is loaded together with the main file. Data load address will be set. Automatic link between main and sub1.
@BRF-LINKER-C<rev> program-file sub2/s2 DATA-FILE main link-to sub1 load ... exit	As for sub1, but in addition there is a LINK-TO between SUB1 and SUB2.

- PREFIX-BRF <file name>,<first unit>,<last unit>,
  <old prefix>,<new prefix>

Editing command.

This command changes a set of names in a BRF code file identified by <file name>. The symbols to be changed are located between <first unit> and <last unit>, starting with the <first unit> and including every unit up to and including the <last unit>. All symbol definition in this range which are prefixed with <old prefix> will be changed.

The symbol names will be changed from

<old prefix><rest of name> to <new prefix><rest of name>.

If no <first unit> is specified, it means the first unit in the file.

If no <last unit> is specified, it means the last unit in the file.

If no <old prefix> is specified, all units are changed.

Example:

```
@BRF-LINKER-C<rev>
- BRF Linker -  10721C<rev>
Brl: list-brf-entries libfile,,
BRF unit: Library mode. Size:   40 P   0 D     INBYTE.......0 P
BRF unit: Library mode. Size:  140 P   0 D     OUTBYTE......0 P
BRF unit: Library mode. Size:  210 P   0 D     OPENFIL......0 P
BRF unit: Library mode. Size:    5 P   0 D     MON1.........0 P
BRF unit: Library mode. Size:    7 P   0 D     MON2.........0 P
BRF unit: Library mode. Size:  110 P   0 D     MON50........0 P

Brl: PREFIX-BRF LIBFILE,MON1,MON50,MON,MY
MON1.....MY1       MON2.....MY2       MON50....MY50

Brl: list-brf-entries libfile,,
BRF unit: Library mode. Size:   40 P   0 D     INBYTE.......0 P
BRF unit: Library mode. Size:  140 P   0 D     OUTBYTE......0 P
BRF unit: Library mode. Size:  210 P   0 D     OPENFIL......0 P
BRF unit: Library mode. Size:    5 P   0 D     MY1..........0 P
BRF unit: Library mode. Size:    7 P   0 D     MY2..........0 P
BRF unit: Library mode. Size:  110 P   0 D     MY50.........0 P
Brl: exit
```

3 INSTALLATION PROCEDURE

Enter the directory on the floppy and copy the file to user SYSTEM:

```
@COPY-FILE
DESTINATION FILE: "BRF-LINKER-C<rev>:PROG"
SOURCE FILE: (210721C:FLOPPY-USER)BRF-LINKER-C<rev>:PROG
```

For ND-100 computers with commercial instruction set (not ND-10), the speed of the BRF-Linker can be increased by using the SYMBOLIC DEBUGGER for ND-100 (version D or later). This must be done before the BRF-Linker is dumped as a reentrant subsystem.

```
@DEBUGGER
PLACE BRF-LINKER-C<rev>,W
STACK-INSTRUCTIONS
EXIT
```

For SINTRAN III version I or later the BRF-Linker are dumped as a reentrant subsystem using the SINTRAN command:

@DUMP-PROGRAM-REENTRANT BRF-LINKER-C<rev>,BRF-LINKER-C<rev>:PROG

For earlier versions of SINTRAN III a BPUN file must be made before dumping the BRF-Linker as a reentrant SUBSYSTEM.
Make the file BRF-LINKER-C<rev>:BPUN by using the subsystem DITAP:

@DITAP "BRF-LINKER-C<rev>:BPUN",BRF-LINKER-C<rev>:PROG

@DUMP-REENTRANT BRF-LINKER-C<rev>,27226,27226,BRF-LINKER-C<rev>:BPUN

NOTE:
If you create the BPUN file, the file BRF-LINKER-C<rev>:PROG must not be deleted from user SYSTEM after the reentrant subsystem is generated.

The BRF-Linker must have a terminal background segment of 128Kw. The background segment size can be changed by the SINTRAN III command:

@CHANGE-BACKGROUND-SEGMENT-SIZE <terminal number>,128