

**NAME**

`ld` – NS16000 link editor (nmeld for cross-support)

**SYNOPSIS**

`ld` [ option ] ... *module* ...

`nmeld` [ option ] ... *module* ...

**DESCRIPTION**

`Ld` combines several object modules into one and resolves external references. In the simplest case several object *modules* are given, and `ld` combines them, producing an *a.out* file which is ready for execution. The output of `ld` is left on the file *a.out*. This file is deleted if errors occurred during the load.

The argument modules are concatenated in the order specified. The entry point of the output is the symbol **start** (unless the `-e` option is specified). Module zero is unused.

The resulting image consists of the module table, followed by each module's program and link table. The link table for a module begins after the program segment for that module. Finally, the static base segment for each module is concatenated at the end of the image. Each segment begins on a 32 bit boundary except for the first static base segment, which begins on a 1024 byte boundary if the user has not specified the `-p` option.

If common symbols are left unresolved, `ld` exports them and allocates space for them. These symbols are defined in the `%initfun` module (see below). Common symbols are aligned on 32 bit boundaries.

Symbols with the special name `%initfun` are coalesced together into one `%initfun` module. `Ld` makes each unique by appending relative mod number and creating a dummy module containing calls to all `%initfun` entry points. This feature is currently used by the compilers for initializing run-time addresses.

The symbols `_etext`, `_edata`, and `_end`, are defined as follows: `_etext` is the first address past the text segment; `_edata` is the first address past the last static base area; `_end` is the first address past the end of the bss area.

`Ld` understands several options.

**-R** *address module*

Exclude program segment of *module* from image. `-R` assumes that *module* is already in shared memory or is present in ROM at *address*. Program relative exports are computed accordingly.

**-T** *base-address*

Load entire image at *base-address*. (Note: This option is only supported on the DB16000 Development Board.)

**-0 -1 -2 -3**

Leave this number of blank module table entries at the beginning of the table in the *a.out* file. Default is 2.

**-w**

Disable warnings. Presently, warnings are issued when a static base relative symbol is imported as an external procedure descriptor.

**-e** *symbol*

Obtain the transfer address from the value of *symbol*.

**-v**

List the utilities `ld` calls and their arguments on standard output. Information produced by the verbose `-v` flag is useful for debugging.

**-o** *filename*

Use *filename* instead of *a.out* for output.

**-m**

Print a load map on the file **ldmap**.

**-mmapfilename**

Print a load map on the file *mapfilename* instead of on **ldmap**.

- M** Produce a primitive load map on standard output, listing the names of the files which will be loaded.
- p** Do not page-align the static base segment. This saves space if it is going into a ROM.
- r** Allow the output file of *ld* to be “re-linked,” i.e. to be the input to another run of *ld*. *Ld* will not define any common symbols and does not give error messages for undefined symbols.
- libname**  
 Search the library */usr/lib/lib<libname>.a* for all unresolved symbols and load the necessary object files. If no **-e** option is given, then the start address of the image will be the symbol **start**.  
*Ld* searches a library when it encounters the library name in the option line so the placement of the **-I** switch is significant (especially when more than one **-I** is used), and it should probably go at the end of the *ld* command.  
 If an archive file is given to the loader, it assumes that the file is a library.  
 Library searches are multipass operations, continuing until no further symbols can be resolved from that library. All libraries must be archive libraries created with *ranlib(1)* and contain a `__SYMDEF` file as the first file in the library.
- s** Strip the symbol table and string table from the output file.

**FILES**

<i>/usr/lib/lib*.a</i>	libraries
<i>a.out</i>	default output file

**SEE ALSO**

*as(1)*, *ar(1)*, *cc(1)*, *ranlib(1)*

**CROSS-SUPPORT**

In a cross-support environment, *ld* is called *nmeld*, and the default output file is *a16.out*. **Nmeld** locates libraries under the **-I** flag in */usr/NSC/lib*. The file that *nmeld* builds is not executable.

**BUGS**

Many flags that are defined in *ld* in the Berkeley version of the command are not yet supported fully. Some of these flags will be supported in the future. Unsupported flags are currently ignored, but cause a message to be printed out explaining that the features are not implemented. These flags are: **-S -x -X -y -z -n -N -t -A -D -d**.