NAME

ld − link editor

SYNOPSIS

ld [**−sulxrdnimX**] [**−K** [p]] [**−o** name] [**−T** symbol] [**−V** version] file ...

DESCRIPTION

Ld combines several object programs into one; resolves external references; and searches libraries (as created by *ar*(1)). In the simplest case several object *files* are given, and *ld* combines them, producing an object module which can be either executed or become the input for a further *ld* run. (In the latter case, the **−r** option must be given to preserve the relocation bits.) The output of *ld* is left on **a.out.** This file is made executable if no errors occurred during the load and the **−r** flag was not specified.

The argument routines are concatenated in the order specified. The entry point of the output is the beginning of the first routine.

If any argument is a library, it is searched exactly once at the point it is encountered in the argument list. Only those routines defining an unresolved external reference are loaded. If a routine from a library references another routine in the library, the referenced routine must appear after the referencing routine in the library. Thus the order of programs within libraries is important.

The symbols **_etext, _etext1, _etext2, _etext3, _edata, _end, _com_mmr** and **mmtble** (**etext, etext1, etext2, etext3, edata, end** and **com_mmr** in C and **mmtble** in assembler) are reserved, and should not be used. If any of the reserved symbols are referred to, the value is the first location above the text, the size of the second switchable text area, the size of the third switchable text area, the size of the fourth switchable text area, the first location above initialized data, the first location above all data, the number of common memory management registers, and the location of the mmtble in data space respectively. It is erroneous to define these symbols.

Ld understands several flag arguments which are written preceded by a −. Except for **−l**, they should appear before the file names.

−s 'Strip' the output, that is, remove the symbol table and relocation bits to save space (but impair the usefulness of the debugger). This information can also be removed by *strip*(1). This option is turned off if there are any undefined symbols.

−u Take the following argument as a symbol and enter it as undefined in the symbol table. This is useful for loading wholly from a library, since initially the symbol table is empty and an unresolved reference is needed to force the loading of the first routine.

−l This option is an abbreviation for a library name. The option **−l** alone stands for **/lib/libc.a**, which is the standard system library for C and assembly language programs. Option **−l***x* stands for **/lib/lib***x***.a,** where *x* is a string. If that does not exist, *ld* tries **/usr/lib/lib***x***.a** A library is searched when its name is encountered, so the placement of a **−l** is significant.

−x Do not preserve local (non-**.globl**) symbols in the output symbol table; only enter external symbols. This option saves some space in the output file.

−X Save local symbols except for those whose names begin with 'L'. This option is used by *cc* to discard internally generated labels while retaining symbols local to routines.

−r Generate relocation bits in the output file so that it can be the subject of another *ld* run. This flag also prevents final definitions from being given to common symbols, and suppresses the 'undefined symbol' diagnostics. Successful completion of *ld* with this option is accompanied by an exit status of 1.

−d Force definition of common storage even if the **−r** flag is present.

−n Arrange that when the output file is executed, the text portion will be read-only and shared among all users executing the file. This involves moving the data areas up to the first possible 4K word boundary following the end of the text.

−i When the output file is executed, the program text and data areas will live in separate address spaces. The only difference between this option and **−n** is that here the data starts at location 0. This option turns off the **−n** option.

−K option is used to build **UNIX** with a text size larger than 65536 bytes. The optional *p* parameter is a number from 1 through 7 indicating how many memory management registers are not changed when switching to an alternate text area. This option along with the order of the object files in lib1.a and lib2.a gives the programmer some control of what routines are in which switchable text area. When building a **UNIX** with switchable text areas, the first object file name in each switchable text area is printed. The default value of p is 7. This switch is good for building the operating system only!

−o The *name* argument after **−o** is used as the name of the *ld* output file, instead of **a.out.**

−m The names of all files and archive members used to create the output file are written to the standard output.

−T The *symbol* argument after **−T** is flagged; up to sixteen symbol names may be specified. Each reference to a flagged symbol is indicated by a printed line containing the type of reference and the referencing module. The symbol name must match exactly, including any initial "_". This option is useful in debugging unreferenced or multiply referenced symbols encountered during the link edit process.

−V The octal number supplied as the *version* argument after **−V** is entered in the header of the output file to indicate the operating system environment expected by the module on execution. **Cc**(1) and **occ**(1) will automatically pass the correct version flag when invoking the loader to specify CB/UNIX Release 2.0 and Release 1.0, respectively. The version specified should be compatible with the libraries used to load the module. See **stamp**(1) for further details.

FILES

/lib/lib?.a	libraries
/usr/lib/lib?.a	more libraries
a.out	output file

SEE ALSO

as(1), ar(1), cc(1), nm(1), occ(1), stamp(1), a.out(5)

DIAGNOSTICS

The message "FINDSYM FAILED TO FIND SYMBOL" usually results from an object file in a switchable text area that has a static routine name or a label that is not .globl if assembler code.