**Study ld command in UNIX / Linux**

**NAME**

ld - the GNU linker

**SYNOPSIS**

**ld**

[**-o** *output* ] *objfile* ...

[**-A** *architecture* ] [**-b** *input-format* ] [**-Bstatic**] [**-Bdynamic**] [**-Bsymbolic**] [**-c** *commandfile* ] [**--cref**] [**-d**|**-dc**|**-dp** ]   
[**-defsym** *symbol* = *expression* ] [**-e** *entry* ] [**-embedded-relocs**] [**-E**] [**-export-dynamic**] [**-f** *name* ] [**--auxiliary** *name* ] [**-F** *name* ] [**--filter** *name* ] [**-format** *input-format* ] [**-g**] [**-G** *size* ] [**-h** *name* ] [**-soname** *name* ] [**--help**] [**-i**] [**-l** *ar* ] [**-L** *searchdir* ] [**-M**] [**-Map** *mapfile* ] [**-m** *emulation* ] [**-n**|**-N**] [**-noinhibit-exec**] [**-no-keep-memory**] [**-oformat** *output-format* ] [**-R** *filename* ] [**-relax**] [**-r**|**-Ur**] [**-rpath** *directory* ] [**-rpath-link** *directory* ] [**-S**] [**-s**] [**-shared**] [**-sort-common**] [**-split-by-reloc** *count* ] [**-split-by-file**] [**-T** *commandfile* ] [**-Ttext** *textorg* ] [**-Tdata** *dataorg* ] [**-Tbss** *bssorg* ] [**-t**] [**-u** *sym* ] [**-V**] [**-v**] [**--verbose**] [**--version**] [**-warn-common**] [**-warn-constructors**] [**-warn-multiple-gp**] [**-warn-once**] [**-warn-section-align**] [**--whole-archive**] [**--no-whole-archive**] [**--wrap** *symbol* ] [**-X**] [**-x**]

**DESCRIPTION**

**ld** combines a number of object and archive files, relocates their data and ties up symbol references. Often the last step in building a new compiled program to run is a call to **ld** .

**ld** accepts Linker Command Language files to provide explicit and total control over the linking process. This man page does not describe the command language; see the `**ld** ' entry in `**info** ', or the manual *ld: the GNU linker* , for full details on the command language and on other aspects of the GNU linker.

This version of **ld** uses the general purpose BFD libraries to operate on object files. This allows **ld** to read, combine, and write object files in many different formats for example, COFF or

**a.out** . Different formats may be linked together to produce any available kind of object file. You can use `**objdump** **-i** ' to get a list of formats supported on various architectures; see **objdump**(**1**).

Aside from its flexibility, the GNU linker is more helpful than other linkers in providing diagnostic information. Many linkers abandon execution immediately upon encountering an error; whenever possible,

**ld** continues executing, allowing you to identify other errors (or, in some cases, to get an output file in spite of the error).

The GNU linker **ld** is meant to cover a broad range of situations, and to be as compatible as possible with other linkers. As a result, you have many choices to control its behavior through the command line, and through environment variables.

**OPTIONS**

The plethora of command-line options may seem intimidating, but in actual practice few of them are used in any particular context. For instance, a frequent use of **ld** is to link standard Unix object files on a standard, supported Unix system. On such a system, to link a file **hello.o** :

$ ld -o output /lib/crt0.o hello.o -lc

This tells **ld** to produce a file called **output** as the result of linking the file **/lib/crt0.o** with **hello.o** and the library **libc.a** which will come from the standard search directories.

The command-line options to **ld** may be specified in any order, and may be repeated at will. For the most part, repeating an option with a different argument will either have no further effect, or override prior occurrences (those further to the left on the command line) of an option.

The exceptions which may meaningfully be used more than once are

**-A** , **-b** (or its synonym **-format** ), **-defsym** ,

**-L** , **-l** , **-R** , and **-u** .

The list of object files to be linked together, shown as *objfile* , may follow, precede, or be mixed in with command-line options; save that an *objfile* argument may not be placed between an option flag and its argument.

Usually the linker is invoked with at least one object file, but other forms of binary input files can also be specified with **-l** ,

**-R** , and the script command language. If *no* binary input files at all are specified, the linker does not produce any output, and issues the message `**No** **input** **files** '.

Option arguments must either follow the option letter without intervening whitespace, or be given as separate arguments immediately following the option that requires them.

**-A***architecture*

In the current release of **ld** , this option is useful only for the Intel 960 family of architectures. In that **ld** configuration, the

*architecture* argument is one of the two-letter names identifying members of the 960 family; the option specifies the desired output target, and warns of any incompatible instructions in the input files. It also modifies the linker's search strategy for archive libraries, to support the use of libraries specific to each particular architecture, by including in the search loop names suffixed with the string identifying the architecture.

For example, if your **ld** command line included `**-ACA** ' as well as `**-ltry** ', the linker would look (in its built-in search paths, and in any paths you specify with **-L** ) for a library with the names

try   
libtry.a   
tryca   
libtryca.a

The first two possibilities would be considered in any event; the last two are due to the use of `**-ACA** '.

Future releases of **ld** may support similar functionality for other architecture families.

You can meaningfully use **-A** more than once on a command line, if an architecture family allows combination of target architectures; each use will add another pair of name variants to search for when **-l** specifies a library.

**-b** *input-format*

Specify the binary format for input object files that follow this option on the command line. You don't usually need to specify this, as

**ld** is configured to expect as a default input format the most usual format on each machine. *input-format* is a text string, the name of a particular format supported by the BFD libraries.

**-format** *input-format*

has the same effect, as does the script command **TARGET**.

You may want to use this option if you are linking files with an unusual binary format. You can also use **-b** to switch formats explicitly (when linking object files of different formats), by including

**-b** *input-format*

before each group of object files in a particular format.

The default format is taken from the environment variable **GNUTARGET** . You can also define the input format from a script, using the command **TARGET** .

**-Bstatic**

Do not link against shared libraries. This is only meaningful on platforms for which shared libraries are supported.

**-Bdynamic**

Link against dynamic libraries. This is only meaningful on platforms for which shared libraries are supported. This option is normally the default on such platforms.

**-Bsymbolic**

When creating a shared library, bind references to global symbols to the definition within the shared library, if any. Normally, it is possible for a program linked against a shared library to override the definition within the shared library. This option is only meaningful on ELF platforms which support shared libraries.

**-c** *commandfile*

Directs **ld** to read link commands from the file

*commandfile* . These commands will completely override **ld** 's default link format (rather than adding to it); *commandfile* must specify everything necessary to describe the target format.

You may also include a script of link commands directly in the command line by bracketing it between `**{** ' and `**}** ' characters.

**--cref**

Output a cross reference table. If a linker map file is being generated, the cross reference table is printed to the map file. Otherwise, it is printed on the standard output.

**-d**

**-dc**

**-dp**

These three options are equivalent; multiple forms are supported for compatibility with other linkers. Use any of them to make **ld** assign space to common symbols even if a relocatable output file is specified (**-r** ). The script command

**FORCE\_COMMON\_ALLOCATION** has the same effect.

**-defsym** *symbol* **=** *expression*

Create a global symbol in the output file, containing the absolute address given by *expression* . You may use this option as many times as necessary to define multiple symbols in the command line. A limited form of arithmetic is supported for the *expression* in this context: you may give a hexadecimal constant or the name of an existing symbol, or use **+** and **-** to add or subtract hexadecimal constants or symbols. If you need more elaborate expressions, consider using the linker command language from a script.

**-e** *entry*

Use *entry* as the explicit symbol for beginning execution of your program, rather than the default entry point. for a discussion of defaults and other ways of specifying the entry point.

**-embedded-relocs**

This option is only meaningful when linking MIPS embedded PIC code, generated by the **-membedded-pic** option to the GNU compiler and assembler. It causes the linker to create a table which may be used at runtime to relocate any data which was statically initialized to pointer values. See the code in testsuite/ld-empic for details.

**-E**

**-export-dynamic**

When creating an ELF file, add all symbols to the dynamic symbol table. Normally, the dynamic symbol table contains only symbols which are used by a dynamic object. This option is needed for some uses of *dlopen.*

**-f** *name*

**--auxiliary** *name*

When creating an ELF shared object, set the internal DT\_AUXILIARY field to the specified name. This tells the dynamic linker that the symbol table of the shared object should be used as an auxiliary filter on the symbol table of the shared object *name.*

**-F** *name*

**--filter** *name*

When creating an ELF shared object, set the internal DT\_FILTER field to the specified name. This tells the dynamic linker that the symbol table of the shared object should be used as a filter on the symbol table of the shared object *name.*

**-format** *input-format*

Synonym for **-b** *input-format* .

**-g**

Accepted, but ignored; provided for compatibility with other tools.

**-G** *size*

Set the maximum size of objects to be optimized using the GP register to *size* under MIPS ECOFF. Ignored for other object file formats.

**-h** *name*

**-soname** *name*

When creating an ELF shared object, set the internal DT\_SONAME field to the specified name. When an executable is linked with a shared object which has a DT\_SONAME field, then when the executable is run the dynamic linker will attempt to load the shared object specified by the DT\_SONAME field rather than the using the file name given to the linker.

**--help**

Print a summary of the command-line options on the standard output and exit. This option and **--version** begin with two dashes instead of one for compatibility with other GNU programs. The other options start with only one dash for compatibility with other linkers.

**-i**

Perform an incremental link (same as option **-r** ).

**-l***ar*

Add an archive file *ar* to the list of files to link. This option may be used any number of times. **ld** will search its path-list for occurrences of **lib** *ar* .a for every *ar* specified.

**-L***searchdir*

This command adds path *searchdir* to the list of paths that

**ld** will search for archive libraries. You may use this option any number of times.

The default set of paths searched (without being specified with

**-L** ) depends on what emulation mode **ld** is using, and in some cases also on how it was configured. The paths can also be specified in a link script with the **SEARCH\_DIR** command.

**-M**

Print (to the standard output file) a link map diagnostic information about where symbols are mapped by **ld** , and information on global common storage allocation.

**-Map** *mapfile*

Print to the file *mapfile* a link map diagnostic information about where symbols are mapped by **ld** , and information on global common storage allocation.

**-m** *emulation*

Emulate the *emulation* linker. You can list the available emulations with the *--verbose* or *-V* options. This option overrides the compiled-in default, which is the system for which you configured **ld**.

**-N**

specifies readable and writable **text** and **data** sections. If the output format supports Unix style magic numbers, the output is marked as **OMAGIC** .

When you use the `**-N** ' option, the linker does not page-align the data segment.

**-n**

sets the text segment to be read only, and **NMAGIC** is written if possible.

**-noinhibit-exec**

Normally, the linker will not produce an output file if it encounters errors during the link process. With this flag, you can specify that you wish the output file retained even after non-fatal errors.

**-no-keep-memory**

The linker normally optimizes for speed over memory usage by caching the symbol tables of input files in memory. This option tells the linker to instead optimize for memory usage, by rereading the symbol tables as necessary. This may be required if the linker runs out of memory space while linking a large executable.

**-o** *output*

*output* is a name for the program produced by **ld** ; if this option is not specified, the name `**a.out** ' is used by default. The script command **OUTPUT** can also specify the output file name.

**-oformat** *output-format*

Specify the binary format for the output object file. You don't usually need to specify this, as

**ld** is configured to produce as a default output format the most usual format on each machine. *output-format* is a text string, the name of a particular format supported by the BFD libraries. The script command **OUTPUT\_FORMAT** can also specify the output format, but this option overrides it.

**-R** *filename*

Read symbol names and their addresses from *filename* , but do not relocate it or include it in the output. This allows your output file to refer symbolically to absolute locations of memory defined in other programs.

**-relax**

An option with machine dependent effects. Currently this option is only supported on the H8/300.

On some platforms, use this option to perform global optimizations that become possible when the linker resolves addressing in your program, such as relaxing address modes and synthesizing new instructions in the output object file.

On platforms where this is not supported, `**-relax** ' is accepted, but has no effect.

**-r**

Generates relocatable output i.e., generate an output file that can in turn serve as input to **ld** . This is often called *partial* linking. As a side effect, in environments that support standard Unix magic numbers, this option also sets the output file's magic number to

**OMAGIC** . If this option is not specified, an absolute file is produced. When linking C++ programs, this option *will* *not* resolve references to constructors; **-Ur** is an alternative.

This option does the same as **-i** .

**-rpath *directory***

Add a directory to the runtime library search path. This is used when linking an ELF executable with shared objects. All **-rpath** arguments are concatenated and passed to the runtime linker, which uses them to locate shared objects at runtime. The **-rpath** option is also used when locating shared objects which are needed by shared objects explicitly included in the link; see the description of the **-rpath-link** option. If **-rpath** is not used when linking an ELF executable, the contents of the environment variable **LD\_RUN\_PATH** will be used if it is defined.

The **-rpath** option may also be used on SunOS. By default, on SunOS, the linker will form a runtime search patch out of all the **-L** options it is given. If a **-rpath** option is used, the runtime search path will be formed exclusively using the **-rpath** options, ignoring the **-L** options. This can be useful when using gcc, which adds many **-L** options which may be on NFS mounted filesystems.

**-rpath-link *directory***

When using ELF or SunOS, one shared library may require another. This happens when an **ld -shared** link includes a shared library as one of the input files.

When the linker encounters such a dependency when doing a non-shared, non-relocateable link, it will automatically try to locate the required shared library and include it in the link, if it is not included explicitly. In such a case, the **-rpath-link** option specifies the first set of directories to search. The **-rpath-link** option may specify a sequence of directory names either by specifying a list of names separated by colons, or by appearing multiple times.

If the required shared library is not found, the linker will issue a warning and continue with the link.

**-S**

Omits debugger symbol information (but not all symbols) from the output file.

**-s**

Omits all symbol information from the output file.

**-shared**

Create a shared library. This is currently only supported on ELF and SunOS platforms (on SunOS it is not required, as the linker will automatically create a shared library when there are undefined symbols and the **-e** option is not used).

**-sort-common**

Normally, when **ld** places the global common symbols in the appropriate output sections, it sorts them by size. First come all the one byte symbols, then all the two bytes, then all the four bytes, and then everything else. This is to prevent gaps between symbols due to alignment constraints. This option disables that sorting.

**-split-by-reloc *count***

Trys to creates extra sections in the output file so that no single output section in the file contains more than *count* relocations. This is useful when generating huge relocatable for downloading into certain real time kernels with the COFF object file format; since COFF cannot represent more than 65535 relocations in a single section. Note that this will fail to work with object file formats which do not support arbitrary sections. The linker will not split up individual input sections for redistribution, so if a single input section contains more than *count* relocations one output section will contain that many relocations.

**-split-by-file**

Similar to **-split-by-reloc** but creates a new output section for each input file.

**-Tbss** *org*

**-Tdata** *org*

**-Ttext** *org*

Use *org* as the starting address for respectively the

**bss** , **data** , or the **text** segment of the output file.

*textorg* must be a hexadecimal integer.

**-T** *commandfile*

Equivalent to **-c** *commandfile*

; supported for compatibility with other tools.

**-t**

Prints names of input files as **ld** processes them.

**-u** *sym*

Forces *sym* to be entered in the output file as an undefined symbol. This may, for example, trigger linking of additional modules from standard libraries. **-u** may be repeated with different option arguments to enter additional undefined symbols.

**-Ur**

For anything other than C++ programs, this option is equivalent to

**-r** : it generates relocatable output i.e., an output file that can in turn serve as input to **ld** . When linking C++ programs, **-Ur** *will* resolve references to constructors, unlike **-r** .

**--verbose**

Display the version number for **ld** and list the supported emulations. Display which input files can and can not be opened.

**-v,** **-V**

Display the version number for **ld** . The **-V** option also lists the supported emulations.

**--version**

Display the version number for **ld** and exit.

**-warn-common**

Warn when a common symbol is combined with another common symbol or with a symbol definition. Unix linkers allow this somewhat sloppy practice, but linkers on some other operating systems do not. This option allows you to find potential problems from combining global symbols.

**-warn-constructors**

Warn if any global constructors are used. This is only useful for a few object file formats. For formats like COFF or ELF, the linker can not detect the use of global constructors.

**-warn-multiple-gp**

Warn if the output file requires multiple global-pointer values. This option is only meaningful for certain processors, such as the Alpha.

**-warn-once**

Only warn once for each undefined symbol, rather than once per module which refers to it.

**-warn-section-align**

Warn if the address of an output section is changed because of alignment. Typically, the alignment will be set by an input section. The address will only be changed if it not explicitly specified; that is, if the SECTIONS command does not specify a start address for the section.

**--whole-archive**

For each archive mentioned on the command line after the **--whole-archive** option, include every object file in the archive in the link, rather than searching the archive for the required object files. This is normally used to turn an archive file into a shared library, forcing every object to be included in the resulting shared library.

**--no-whole-archive**

Turn off the effect of the **--whole-archive** option for archives which appear later on the command line.

**--wrap** *symbol*

Use a wrapper function for *symbol.* Any undefined reference to *symbol* will be resolved to **\_\_wrap\_***symbol.* Any undefined reference to **\_\_real\_***symbol* will be resolved to *symbol.*

**-X**

Delete all temporary local symbols. For most targets, this is all local symbols whose names begin with `**L** '.

**-x**

Delete all local symbols.

**ENVIRONMENT**

You can change the behavior of **ld** with the environment variable **GNUTARGET** .

**GNUTARGET** determines the input-file object format if you don't use **-b** (or its synonym **-format** ). Its value should be one of the BFD names for an input format. If there is no

**GNUTARGET** in the environment, **ld** uses the natural format of the host. If **GNUTARGET** is set to **default** then BFD attempts to discover the input format by examining binary input files; this method often succeeds, but there are potential ambiguities, since there is no method of ensuring that the magic number used to flag object-file formats is unique. However, the configuration procedure for BFD on each system places the conventional format for that system first in the search-list, so ambiguities are resolved in favor of convention.