

NAME

strip – strip symbol table, debugging and line number information from an object file

SYNOPSIS

strip [-IVx] [-c [!]class1,[!]class2,...] file...

DESCRIPTION

The **strip** command removes sections from **ELF** object files. By default, the symbol table **SHT_SYMTAB** and its associated string table, symbol sort sections, debugging information, and line number information is removed.

The default set of candidate sections corresponds to the **debug** and **symbol** class identifiers to the **-c** option. The **-c**, **-l**, or **-x** options can be used to select a different set of candidate sections.

Once this stripping process has been done, limited symbolic debugging access is available for that file. Therefore, this command is normally run only on production modules that have been debugged and tested.

If **strip** is executed on a common archive file (see **ar.h(3HEAD)**) in addition to processing the members, **strip** removes the archive symbol table. The archive symbol table must be restored by executing the **ar(1)** command with the **-s** option before the archive can be linked by the **ld(1)** command.

strip is used to reduce the file storage overhead taken by the object file.

OPTIONS

The following options are supported:

-c=[!]class1,[!]class2,...

Strip a specific class of section. This option provides fine grained control over the sections to be removed.

The strip class descriptions that follow only apply to non-allocatable sections.

Some classes cause other classes to be implicitly included, or **encapsulated**. Such cases are noted in the class descriptions below. Each class token can be prepended with a **'!'** to indicate that the class should not be removed. This definition can be useful to prevent a class from including another normally encapsulated class. For example, while the **nonalloc** class encapsulates all non-allocable sections, **-c nonalloc,!note** removes all non-allocatable sections except for the note section.

Stripped sections are completely removed from the output object.

The following classes of section can be defined.

nonalloc

Strip any non-allocatable section. These sections are identified as not including the **SHF_ALLOC** section flag. This class encapsulates all of the other classes, except for the **symbol** class. The **nonalloc** class is often sufficient by itself to remove any unwanted sections.

annotate

Strip any annotation section. These sections provide information that is used by memory access tools, and coverage related tools. These sections are identified by having a **SHT_SUNW_ANNOTATE** section type.

comment

Strip any comment section. These sections are identified by having a **.comment** section name. Alternatively, the **mcs(1)** utility is commonly used to manipulate comment sections.

compcom

Strip any compiler commentary section. These sections are identified by having a **.compcom** section name.

ctf

Strip **CTF** (Compact C Type Format) sections. These sections are identified by having a **.SUNW_ctf** section name, and also by having a **SHT_SUNW_ctf** section type.

debug

Strip sections commonly used to contain debugging data. These sections are identified by having a **.debug***, **.stab***, or **.zdebug*** section name. These sections are also identified by having a section type of **SHT_SUNW_DEBUG** or **SHT_SUNW_DEBUGSTR**. This class also encapsulates the **compcom**, and **line** classes.

exclude

Strip any excludable section. These sections are identified by having a **SHF_EXCLUDE** section flag. This class can be useful when manipulating a relocatable object. By default, such sections are automatically excluded when a dynamic object, or kernel module, is created, and are retained when creating a relocatable object.

line

Strip line number debug section. These sections are named **.line**.

note

Strip any note section. These sections are identified by having a **SHT_NOTE** section type.

sort_sym_addr, sort_sym_name**sort, sort_addr, sort_name, sort_sym**

Strip the symbol sort sections associated to the **SHT_SYMTAB** symbol table. Each of these symbol tables typically have associated symbol sort sections, sorted by address, and by name.

sort_sym_addr By-address **.symtab** symbol sort sections (**SHT_SUNW_symsort**, **SHT_SUNW_tlssort**).

sort_sym_name By-name **.symtab** symbol sort sections (**SHT_SUNW_symnsort**).

The remaining symbol sort section class tokens provide combinations of the previous items, and

are provided for convenience.

sort, sort_sym **sort_sym_addr**, and **sort_sym_name**.

sort_addr **sort_sym_addr**.

sort_name **sort_sym_name**.

symbol

Strip any non-allocatable symbol table, providing the file is not a relocatable object. These sections are identified by having a **SHT_SYMTAB** section type. This class also encapsulates the **sort_sym** classes. Any associated string table, symbol sort sections, or **CTF** section are also removed.

-l

Strip line number information only. This option is equivalent to using the **-c** option with the **line** class identifier.

-V

--version

Print version information.

-x

Does not strip the symbol table. This option is equivalent to using the **-c** option with the **!symbol** class identifier.

-?

--help

Print usage message and immediately exit.

OPERANDS

The following operand is supported:

file A path name referring to an executable file.

ENVIRONMENT VARIABLES

See **environ(7)** for descriptions of the following environment variables that affect the execution of **strip**: **LANG**, **LC_ALL**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

FILES

/tmp/strip_* Temporary files

ATTRIBUTES

See **attributes(7)** for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities
Interface Stability	Committed
Standard	See standards(7) .

SEE ALSO

ar(1), **as(1)**, **elfcompress(1)**, **ld(1)**, **mcs(1)**, **tmpnam(3C)**, **elf(3ELF)**, **ar.h(3HEAD)**, **a.out(5)**, **attributes(7)**, **environ(7)**, **standards(7)**

NOTES

The **strip** command is used to remove a standard predefined set of sections from an ELF object file. To remove a user specified section by name, see **mcs(1)**.

The **strip** command is unable to modify core files, or ancillary objects.

The **strip -c** option corresponds to the **ld -z strip-class** option. Performing this operation at link-edit time, rather than using **strip** to post process objects is more efficient, but otherwise equivalent. See **ld(1)**.