-- --

IEEE P1003.2 Draft 11.2 – September 1991

make.

6.3 strip — Remove unnecessary information from executable files

6.3.1 Synopsis

strip *file* . . .

6.3.2 Description

The strip utility shall remove from executable files named by the *file* operands any information the implementor deems unnecessary to proper execution of those files. The nature of that information is unspecified. The effect of strip shall be the same as the use of the −s option to any of the compilers defined by this standard.

6.3.3 Options

None.

6.3.4 Operands

The following operand shall be supported by the implementation:

file A pathname referring to an executable file.

6.3.5 External Influences

6.3.5.1 Standard Input

None.

6.3.5.2 Input Files

The input files shall be in the form of executable files successfully produced by any compiler defined by this standard.

6.3.5.3 Environment Variables

The following environment variables shall affect the execution of strip:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments).

LC_MESSAGES This variable shall determine the language in which messages should be written.

6.3.5.4 Asynchronous Events

Default.

6.3.6 External Effects

6.3.6.1 Standard Output

None.

6.3.6.2 Standard Error

Used only for diagnostic messages.

6.3.6.3 Output Files

The `strip` utility shall produce executable files of unspecified format.

6.3.7 Extended Description

None.

6.3.8 Exit Status

The `strip` utility shall exit with one of the following values:

0 Successful completion.

>0 An error occurred.

6.3.9 Consequences of Errors

Default.

6.3.10 Rationale. *(This subclause is not a part of P1003.2)*

Examples, Usage

None.

History of Decisions Made

Historically, this utility has been used to remove the symbol table from an executable file. It was included since it is known that the amount of symbolic information can amount to several megabytes; the ability to remove it in a portable manner was deemed important, especially for smaller systems.

The behavior of `strip` is said to be the same as the `−s` option to a compiler. While the end result is essentially the same it is not required to be identical. The same effect can be achieved with either `−s` during a compile or a `strip` on the final object file.

Section 7: Language-Independent System Services

This clause contains functional specifications for services that give applications access to features defined elsewhere in this standard. These services allow applications written in high-level languages to

(1) execute commands using the shell language,

(2) obtain values of environment variables,

(3) perform regular expression and pattern matching,

(4) process command arguments in a standard manner,

(5) generate pathnames from a pattern,

(6) perform shell word expansions,

(7) obtain system configuration information, and

(8) set locale control information

This clause does not define interfaces, but services that shall be provided by the interfaces in a language-dependent binding. This clause is optional, in that an implementation is not required to support any language binding to these services. However, any language binding shall support all of the services described here. Implementations therefore provide support for services in this clause by supplying a language-dependent binding such as the one defined in Annex B. Such a system would specify conformance to the language-dependent binding, not to the language-independent bindings given here.

7.0.1 Language-Independent System Services Rationale. *(This subclause is not a part of P1003.2)*

Section 7 essentially is a metastandard, in that it specifies services that must be in a language-dependent binding. An implementation conforms to a specific language-dependent binding such as for the C language, in Annex B, and the language-dependent binding must conform to the specifications in this clause.

In this standard, the language-independent specifications have not yet been developed. The language-independent syntax is being created in parallel by the POSIX. 1 working group. Therefore, the C language bindings temporarily described in Annex B are actually the full interface specifications. It is the intention of the P1003.2 working group to rectify this situation in a later supplement by moving the majority of the interface specifications back into this clause,