

NAME

intro – introduction to library functions

DESCRIPTION

Section 3 of the manual describes all library functions excluding the library functions (system call wrappers) described in Section 2, which implement system calls.

Many of the functions described in the section are part of the Standard C Library (*libc*). Some functions are part of other libraries (e.g., the math library, *libm*, or the real-time library, *librt*) in which case the manual page will indicate the linker option needed to link against the required library (e.g., *-lm* and *-lrt*, respectively, for the aforementioned libraries).

In some cases, the programmer must define a feature test macro in order to obtain the declaration of a function from the header file specified in the man page SYNOPSIS section. (Where required, these *feature test macros* must be defined before including *any* header files.) In such cases, the required macro is described in the man page. For further information on feature test macros, see **feature_test_macros(7)**.

Subsections

Section 3 of this manual is organized into subsections that reflect the complex structure of the standard C library and its many implementations:

- 3const
- 3head
- 3type

This difficult history frequently makes it a poor example to follow in design, implementation, and presentation.

Ideally, a library for the C language is designed such that each header file presents the interface to a coherent software module. It provides a small number of function declarations and exposes only data types and constants that are required for use of those functions. Together, these are termed an API or *application program interface*. Types and constants to be shared among multiple APIs should be placed in header files that declare no functions. This organization permits a C library module to be documented concisely with one header file per manual page. Such an approach improves the readability and accessibility of library documentation, and thereby the usability of the software.

STANDARDS

Certain terms and abbreviations are used to indicate UNIX variants and standards to which calls in this section conform. See **standards(7)**.

NOTES**Authors and copyright conditions**

Look at the header of the manual page source for the author(s) and copyright conditions. Note that these can be different from page to page!

SEE ALSO

intro(2), **errno(3)**, **capabilities(7)**, **credentials(7)**, **environ(7)**, **feature_test_macros(7)**, **libc(7)**, **math_error(7)**, **path_resolution(7)**, **pthreads(7)**, **signal(7)**, **standards(7)**, **system_data_types(7)**