NAME

strtol, **strtoll**, **strtoimax**, **strtoq** - convert a string value to a *long*, *long long*, *intmax_t* or *quad_t* integer

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <stdlib.h>
#include <limits.h>

long
strtol(*const char * restrict nptr, char ** restrict endptr, int base*);

long long
strtoll(*const char * restrict nptr, char ** restrict endptr, int base*);

#include <inttypes.h>

intmax_t
strtoimax(*const char * restrict nptr, char ** restrict endptr, int base*);

#include <sys/types.h>
#include <stdlib.h>
#include <limits.h>

quad_t
strtoq(*const char *nptr, char **endptr, int base*);

DESCRIPTION

The **strtol**() function converts the string in *nptr* to a *long* value. The **strtoll**() function converts the string in *nptr* to a *long long* value. The **strtoimax**() function converts the string in *nptr* to an *intmax_t* value. The **strtoq**() function converts the string in *nptr* to a *quad_t* value. The conversion is done according to the given *base*, which must be between 2 and 36 inclusive, or be the special value 0.

The string may begin with an arbitrary amount of white space (as determined by isspace(3)) followed by a single optional ‘+’ or ‘-’ sign. If *base* is zero or 16, the string may then include a "0b" prefix, and the number will be read in base 2; or it may include a "0x" prefix, and the number will be read in base 16; otherwise, a zero *base* is taken as 10 (decimal) unless the next character is ‘0’, in which case it is taken as 8 (octal).

The remainder of the string is converted to a *long*, *long long*, *intmax_t* or *quad_t* value in the obvious manner, stopping at the first character which is not a valid digit in the given base. (In bases above 10, the letter 'A' in either upper or lower case represents 10, 'B' represents 11, and so forth, with 'Z' representing 35.)

If *endptr* is not NULL, **strtol**() stores the address of the first invalid character in **endptr*. If there were no digits at all, however, **strtol**() stores the original value of *nptr* in **endptr*. (Thus, if **nptr* is not '\0' but ***endptr* is '\0' on return, the entire string was valid.)

RETURN VALUES

The **strtol**(), **strtoll**(), **strtoimax**() and **strtoq**() functions return the result of the conversion, unless the value would underflow or overflow. If no conversion could be performed, 0 is returned and the global variable *errno* is set to EINVAL (the last feature is not portable across all platforms). If an overflow or underflow occurs, *errno* is set to ERANGE and the function return value is clamped according to the following table.

Function	**underflow**	**overflow**
strtol()	LONG_MIN	LONG_MAX
strtoll()	LLONG_MIN	LLONG_MAX
strtoimax()	INTMAX_MIN	INTMAX_MAX
strtoq()	LLONG_MIN	LLONG_MAX

ERRORS

[EINVAL] The value of *base* is not supported or no conversion could be performed (the last feature is not portable across all platforms).

[ERANGE] The given string was out of range; the value converted has been clamped.

SEE ALSO

atof(3), atoi(3), atol(3), strtod(3), strtonum(3), strtoul(3), wcstol(3)

STANDARDS

The **strtol**() function conforms to ISO/IEC 9899:1990 ("ISO C90"). The **strtoll**() and **strtoimax**() functions conform to ISO/IEC 9899:1999 ("ISO C99"). The BSD **strtoq**() function is deprecated.