Fondamenti di Informatica II e Lab.

La libreria <stdint.h>

Pagine estratte dal sito www.cplusplus.com

Ultimo aggiornamento 28/10/2016

header

<cstdint> (stdint.h)

Integer types

This header defines a set of integral type aliases with specific width requirements, along with macros specifying their limits and macro functions to create values of these types.

Types

The following are typedefs of *fundamental integral types* or *extended integral types.*

signed type	unsigned type	description
intmax_t	uintmax_t	Integer type with the maximum width supported.
int8_t	uint8_t	Integer type with a width of exactly 8, 16, 32, or 64 bits. For signed types, negative values are represented using 2's complement. No padding bits. **Optional:** These typedefs are not defined if no types with such characteristics exist.*
int16_t	uint16_t	
int32_t	uint32_t	
int64_t	uint64_t	
int_least8_t	uint_least8_t	Integer type with a minimum of 8, 16, 32, or 64 bits. No other integer type exists with lesser size and at least the specified width.
int_least16_t	uint_least16_t	
int_least32_t	uint_least32_t	
int_least64_t	uint_least64_t	
int_fast8_t	uint_fast8_t	Integer type with a minimum of 8, 16, 32, or 64 bits. At least as fast as any other integer type with at least the specified width.
int_fast16_t	uint_fast16_t	
int_fast32_t	uint_fast32_t	
int_fast64_t	uint_fast64_t	
intptr_t	uintptr_t	Integer type capable of holding a value converted from a void pointer and then be converted back to that type with a value that compares equal to the original pointer. **Optional:** These typedefs may not be defined in some library implementations.*

Some of these typedefs may denote the same types. Therefore, function overloads should not rely on these being different.

* Notice that some types are optional (and thus, with no portability guarantees). A particular library implementation may also define additional types with other widths supported by its system. In any case, if either the signed or the unsigned version is defined, both the signed and unsigned versions are defined.

Macros

Limits of cstdint types

Macro	description	defined as
INTMAX_MIN	Minimum value of intmax_t	$-(2^{63}-1)$, or lower
INTMAX_MAX	Maximum value of intmax_t	$2^{63}-1$, or higher
UINTMAX_MAX	Maximum value of uintmax_t	$2^{64}-1$, or higher
INT*N*_MIN	Minimum value of exact-width signed type	Exactly $-2^{(N-1)}$
INT*N*_MAX	Maximum value of exact-width signed type	Exactly $2^{(N-1)}-1$
UINT*N*_MAX	Maximum value of exact-width unsigned type	Exactly $2^{N}-1$
INT_LEAST*N*_MIN	Minimum value of minimum-width signed type	$-(2^{(N-1)}-1)$, or lower
INT_LEAST*N*_MAX	Maximum value of minimum-width signed type	$2^{(N-1)}-1$, or higher
UINT_LEAST*N*_MAX	Maximum value of minimum-width unsigned type	$2^{N}-1$, or higher
INT_FAST*N*_MIN	Minimum value of fastest minimum-width signed type	$-(2^{(N-1)}-1)$, or lower
INT_FAST*N*_MAX	Maximum value of fastest minimum-width signed type	$2^{(N-1)}-1$, or higher
UINT_FAST*N*_MAX	Maximum value of fastest minimum-width unsigned type	$2^{N}-1$, or higher
INTPTR_MIN	Minimum value of intptr_t	$-(2^{15}-1)$, or lower
INTPTR_MAX	Maximum value of intptr_t	$2^{15}-1$, or higher
UINTPTR_MAX	Maximum value of uintptr_t	$2^{16}-1$, or higher

Where N is one in 8, 16, 32, 64, or any other type width supported by the library.

Only the macros corresponding to types supported by the library are defined.

Limits of other types

Limits of other standard integral types:

Macro	description	defined as
SIZE_MAX	Maximum value of size_t	$2^{64}-1$, or higher
PTRDIFF_MIN	Minimum value of ptrdiff_t	$-(2^{16}-1)$, or lower
PTRDIFF_MAX	Maximum value of ptrdiff_t	$2^{16}-1$, or higher
SIG_ATOMIC_MIN	Minimum value of sig_atomic_t	if sig_atomic_t is signed: -127, or lower if sig_atomic_t is unsigned: 0
SIG_ATOMIC_MAX	Maximum value of sig_atomic_t	if sig_atomic_t is signed: 127, or higher if sig_atomic_t is unsigned: 255, or higher
WCHAR_MIN	Minimum value of wchar_t	if wchar_t is signed: -127, or lower if wchar_t is unsigned: 0
WCHAR_MAX	Maximum value of wchar_t	if wchar_t is signed: 127, or higher if wchar_t is unsigned: 255, or higher
WINT_MIN	Minimum value of wint_t	if wint_t is signed: -32767, or lower if wint_t is unsigned: 0
WINT_MAX	Maximum value of wint_t	if wint_t is signed: 32767, or higher if wint_t is unsigned: 65535, or higher

Function-like macros

These function-like macros expand to integer constants suitable to initialize objects of the types above:

Macro	description
INTMAX_C	expands to a value of type intmax_t
UINTMAX_C	expands to a value of type uintmax_t
INT*N*_C	expands to a value of type int_least*N*_t
UINT*N*_C	expands to a value of type uint_least*N*_t

For example:

```
INTMAX_C(2012)  // expands to 2012LL or similar
```