

N3349: Abs Without Undefined Behavior

Lenard Mollenkopf*, Anton Zellerhoff†

September 18, 2024

Document: N3349

Audience: WG14

Proposal category: New feature

Document hosted at <https://www.overleaf.com/read/yvbsgssqndfd#24e519>

Abstract The `abs` functions have undefined behaviour, which can be avoided, if `unsigned` return types are used.

*Unaffiliated, wg14@lenardmollenkopf.de

†Unaffiliated, wg14@ascz.de

1 Introduction and Rationale

The functions `abs`, `labs`, `llabs` and `imaxabs` all result in undefined behaviour, if they are invoked with the smallest number representable in their corresponding type. As the absolute value of this number can not be represented in their corresponding type.

This edge case is avoidable if the unsigned version of the type is used for the return type instead, as it can represent values up to $2^n - 1$, opposed to $2^{n-1} - 1$ that the signed type can represent.

Adding such functions that make use of the unsigned type would simplify the development of programs that do not invoke undefined behaviour.

2 Summary of proposed changes

Add the functions `uabs`, `ulabs`, `ullabs` to `stdlib.h` and `uimaxabs` to `inttypes.h`.

3 Proposal

The following functions should be added to `stdlib.h`, each computing the absolute value of `j`:

```
unsigned int uabs(int j);
unsigned long int ulabs(long int j);
unsigned long long int ullabs(long long int j);
```

The following function should be added to `inttypes.h`, computing the absolute value of `j`:

```
uintmax_t uimaxabs(intmax_t j);
```

4 Impact

The introduction of the functions will conflict with existing code already using those names. In most cases functions with the same name will probably do exactly the same as described in this proposal.

Using a library version-test macro can help with backwards compatible code. We do not see the need of a dedicated test macro `__has_c_uabs`.

5 Proposed wording

We propose to modify sub-section 7.24.7.1

Synopsis

7.24.7.1 The `abs`, `labs`, ~~and~~ `llabs`, `uabs`, `ulabs`, and `ullabs` functions

```
#include <stdlib.h>
int abs(int j);
long int labs(long int j);
long long int llabs(long long int j);
unsigned int uabs(int j);
unsigned long int ulabs(long int j);
unsigned long long int ullabs(long long int j);
```

Description

The `abs`, `labs`, ~~and~~ `llabs`, `uabs`, `ulabs`, and `ullabs` functions compute the absolute value of an integer `j`. If the result cannot be represented, the behavior is undefined.³⁵³⁾

and the footnote 353

353) Only the absolute value of the most negative number is not representable for `abs`, `labs`, and `llabs`.

And likewise sub-section 7.8.3.1

Synopsis

7.8.3.1 The `imaxabs` and `uimaxabs` functions

```
#include <inttypes.h>
intmax_t imaxabs(intmax_t j);
uintmax_t uimaxabs(intmax_t j);
```

Description

The `imaxabs` and `uimaxabs` functions compute the absolute value of an integer `j`. If the result cannot be represented, the behavior is undefined.²⁵⁵⁾

and footnote 255

255) Only the absolute value of the most negative number is not representable for `imaxabs`.

6 Possible Implementation

Excerpt:

```
#include <stdckdint.h>

unsigned int uabs (int j) {
    if(j < 0){
        unsigned int u;
        ckd_mul(&u,j,-1);
        return u;
    }
    return j;
}
```

Complete implementation at <https://godbolt.org/z/63q1TrM9b>.