DD. Unspecified Behavior

Created by Martin Sebor, last modified by Will Snavely on Feb 10, 2016

According to the C Standard, Annex J, J.1 [ISO/IEC 9899:2011], the behavior of a program is unspecified in the circumstances outlined the following table. The descriptions of unspecified behaviors in the "Description" column are direct quotes from the standard. The parenthesized numbers refer to the subclause of the C Standard (C11) that identifies the unspecified behavior. The "Guideline" column in the table identifies the coding practices that address the specific case of unspecified behavior (USB).

USB	Description	Guideline
1	The manner and timing of static initialization (5.1.2).	
2	The termination status returned to the hosted environment if the return type of `main` is not compatible with `int` (5.1.2.2.3).	
3	The values of objects that are neither lock-free atomic objects nor of type `volatile sig_atomic_t` and the state of the floating-point environment when the processing of the abstract machine is interrupted by receipt of a signal (5.1.2.3).	
4	The behavior of the display device if a printing character is written when the active position is at the final position of a line (5.2.2).	
5	The behavior of the display device if a backspace character is written when the active position is at the initial position of a line 5.2.2).	
6	The behavior of the display device if a horizontal tab character is written when the active position is at or past the last defined horizontal tabulation position (5.2.2).	
7	The behavior of the display device if a vertical tab character is written when the active position is at or past the last defined vertical tabulation position (5.2.2).	
8	How an extended source character that does not correspond to a universal character name counts toward the significant initial characters in an external identifier (5.2.4.1).	
9	Many aspects of the representations of types (6.2.6).	
10	The value of padding bytes when storing values in structures or unions (6.2.6.1).	
11	The values of bytes that correspond to union members other than the one last stored into (6.2.6.1).	EXP39-C
12	The representation used when storing a value in an object that has more than one object representation for that value (6.2.6.1).	
13	The values of any padding bits in integer representations (6.2.6.2).	
14	Whether certain operators can generate negative zeros and whether a negative zero becomes a normal zero when stored in an object (6.2.6.2).	

USB	Description	Guideline
16	The order in which subexpressions are evaluated and the order in which side effects take place, except as specified for the function-call `()`, `&&`, `||`, `?:`, and comma operators (6.5).	EXP30-C
17	The order in which the function designator, arguments, and subexpressions within the arguments are evaluated in a function call (6.5.2.2).	
18	The order of side effects among compound literal initialization list expressions (6.5.2.5).	
19	The order in which the operands of an assignment operator are evaluated (6.5.16).	
20	The alignment of the addressable storage unit allocated to hold a bit-field (6.7.2.1).	
21	Whether a call to an inline function uses the inline definition or the external definition of the function (6.7.4).	
22	Whether or not a size expression is evaluated when it is part of the operand of a `sizeof` operator and changing the value of the size expression would not affect the result of the operator (6.7.6.2).	EXP44-C
23	The order in which any side effects occur among the initialization list expressions in an initializer (6.7.9).	
24	The layout of storage for function parameters (6.9.1).	
25	When a fully expanded macro replacement list contains a function-like macro name as its last preprocessing token and the next preprocessing token from the source file is a `(`, and the fully expanded replacement of that macro ends with the name of the first macro and the next preprocessing token from the source file is again a `(`, whether that is considered a nested replacement (6.10.3).	
26	The order in which `#` and `##` operations are evaluated during macro substitution (6.10.3.2, 6.10.3.3).	
27	The state of the floating-point status flags when execution passes from a part of the program translated with `FENV_ACCESS` "off" to a part translated with `FENV_ACCESS` "on" (7.6.1).	
28	The order in which `feraiseexcept` raises floating-point exceptions, except as stated in F.8.6 (7.6.2.3).	
29	Whether `math_errhandling` is a macro or an identifier with external linkage (7.12).	DCL37-C
30	The results of the `frexp` functions when the specified value is not a floating-point number (7.12.6.4).	
31	The numeric result of the `ilogb` functions when the correct value is outside the range of the return type (7.12.6.5, F.10.3.5).	
32	The result of rounding when the value is out of range (7.12.9.5, 7.12.9.7, F.10.6.5).	
33	The value stored by the `remquo` functions in the object pointed to by `quo` when `y` is zero (7.12.10.3).	
34	Whether a comparison macro argument that is represented in a format wider than its semantic type is converted to the semantic type (7.12.14).	

USB	Description	Guideline
36	Whether `va_copy` and `va_end` are macros or identifiers with external linkage (7.16.1).	DCL37-C
37	The hexadecimal digit before the decimal point when a non-normalized floating-point number is printed with an `a` or `A` conversion specifier (7.21.6.1, 7.29.2.1).	
38	The value of the file position indicator after a successful call to the `ungetc` function for a text stream, or the `ungetwc` function for any stream, until all pushed-back characters are read or discarded (7.21.7.10, 7.29.3.10).	
39	The details of the value stored by the `fgetpos` function (7.21.9.1).	
40	The details of the value returned by the `ftell` function for a text stream (7.21.9.4).	
41	Whether the `strtod`, `strtof`, `strtold`, `wcstod`, `wcstof`, and `wcstold` functions convert a minus-signed sequence to a negative number directly or by negating the value resulting from converting the corresponding unsigned sequence (7.22.1.3, 7.29.4.1.1).	
42	The order and contiguity of storage allocated by successive calls to the `calloc`, `malloc`, and `realloc` functions (7.22.3).	
43	The amount of storage allocated by a successful call to the `calloc`, `malloc`, and `realloc` function when 0 bytes was requested (7.22.3).	MEM04-C
44	Whether a call to the `atexit` function that does not happen before the `exit` function is called will succeed (7.22.4.2).	
45	Whether a call to the `at_quick_exit` function that does not happen before the `quick_exit` function is called will succeed (7.22.4.3).	
46	Which of two elements that compare as equal is matched by the `bsearch` function (7.22.5.1).	
47	The order of two elements that compare as equal in an array sorted by the `qsort` function (7.22.5.2).	
48	The encoding of the calendar time returned by the `time` function (7.27.2.4).	MSC05-C
49	The characters stored by the `strftime` or `wcsftime` function if any of the time values being converted is outside the normal range (7.27.3.5, 7.29.5.1).	
50	Whether an encoding error occurs if a `wchar_t` value that does not correspond to a member of the extended character set appears in the format string for a function in 7.29.2 or 7.29.5 and the specified semantics do not require that value to be processed by `wcrtomb` (7.29.1).	
51	The conversion state after an encoding error occurs (7.28.1.1, 7.28.1.2, 7.28.1.3, 7.28.1.4, 7.29.6.3.2, 7.29.6.3.3, 7.29.6.4.1, 7.29.6.4.2).	
52	The resulting value when the "invalid" floating-point exception is raised during IEC 60559 floating to integer conversion (F.4).	
53	Whether conversion of non-integer IEC 60559 floating values to integer raises the "inexact" floating-point exception (F.4).	
54	Whether or when library functions in `<math.h>` raise the "inexact" floating-point exception in an IEC 60559 conformant implementation (F.10).	
55	Whether or when library functions in `<math.h>` raise an undeserved "underflow" floating-point exception in an IEC 60559 conformant implementation (F.10).	

USB	Description	Guideline
57	The numeric result returned by the `lrint`, `llrint`, `lround`, and `llround` functions if the rounded value is outside the range of the return type (F.10.6.5, F.10.6.7).	
58	The sign of one part of the complex result of several math functions for certain exceptional values in IEC 60559 compatible implementations (G.6.1.1, G.6.2.2, G.6.2.3, G.6.2.4, G.6.2.5, G.6.2.6, G.6.3.1, G.6.4.2).	

← ↑

bm section

©2020 Carnegie Mellon University