Floating-Point Status Flags and Control Modes

Section 7.2 discussed some of the basic properties of IEEE Standard 754. Section 23.4, which covered the C99 additions to the <math.h> header, gave additional detail. Some of that discussion, particularly concerning exceptions and rounding directions, is directly relevant to the <fenv.h> header. Before we continue, let's review some of the material from Section 23.4 as well as define a few new terms.

A ***floating-point status flag*** is a system variable that's set when a floating-point exception is raised. In the IEEE standard, there are five types of floating-point exceptions: *overflow*, *underflow*, *division by zero*, *invalid operation* (the result of an arithmetic operation was NaN), and *inexact* (the result of an arithmetic operation had to be rounded). Each exception has a corresponding status flag.

The <fenv.h> header declares a type named fexcept_t that's used for working with the floating-point status flags. An fexcept_t object represents the collective value of these flags. Although fexcept_t can simply be an integer type, with single bits representing individual flags, the C99 standard doesn't make this a requirement. Other alternatives exist, including the possibility that fexcept_t is a structure, with one member for each exception. This member could store additional information about the corresponding exception, such as the address of the floating-point instruction that caused the exception to be raised.

A ***floating-point control mode*** is a system variable that may be set by a program to change the future behavior of floating-point arithmetic. The IEEE standard requires a "directed-rounding" mode that controls the rounding direction when a number can't be represented exactly using a floating-point representation. There are four rounding directions: (1) *Round toward nearest.* Rounds to the nearest representable value. If a number falls halfway between two values, it's rounded to the "even" value (the one whose least significant bit is zero). (2) *Round toward zero.* (3) *Round toward positive infinity.* (4) *Round toward negative infinity.* The default rounding direction is round toward nearest. Some implementations of the IEEE standard provide two additional control modes: a mode that controls rounding precision and a "trap enablement" mode that determines whether a floating-point processor will trap (or stop) when an exception is raised.

The term ***floating-point environment*** refers to the combination of floating-point status flags and control modes supported by a particular implementation. A value of type fenv_t represents an entire floating-point environment. The fenv_t type, like the fexcept_t type, is declared in <fenv.h>.

<fenv.h> Macros

The <fenv.h> header potentially defines the macros listed in Table 27.8. Only two of these macros (FE_ALL_EXCEPT and FE_DFL_ENV) are required, however. An implementation may define additional macros not listed in the table; the names of these macros must begin with FE_ and an uppercase letter.