NAME

fpgetround, **fpsetround**, **fpsetprec**, **fpgetprec**, **fpgetmask**, **fpsetmask**, **fpgetsticky**, **fpresetsticky** - IEEE floating point interface

SYNOPSIS

#include <ieeefp.h>

```
typedef enum {
        FP_RN,          /* round to nearest */
        FP_RM,          /* round down to minus infinity */
        FP_RP,          /* round up to plus infinity */
        FP_RZ           /* truncate */
} fp_rnd_t;
```

fp_rnd_t
fpgetround(*void*);

fp_rnd_t
fpsetround(*fp_rnd_t direction*);

```
typedef enum {
        FP_PS,          /* 24 bit (single-precision) */
        FP_PRS,         /* reserved */
        FP_PD,          /* 53 bit (double-precision) */
        FP_PE           /* 64 bit (extended-precision) */
} fp_prec_t;
```

fp_prec_t
fpgetprec(*void*);

fp_prec_t
fpsetprec(*fp_prec_t precision*);

```
#define fp_except_tint
#define FP_X_INV 0x01   /* invalid operation */
#define FP_X_DNML       0x02    /* denormal */
#define FP_X_DZ         0x04    /* zero divide */
#define FP_X_OFL0x08    /* overflow */
#define FP_X_UFL0x10    /* underflow */
#define FP_X_IMP 0x20   /* (im)precision */
#define FP_X_STK0x40    /* stack fault */
```

fp_except_t

fpgetmask(*void*);

fp_except_t
fpsetmask(*fp_except_t mask*);

fp_except_t
fpgetsticky(*void*);

fp_except_t
fpresetsticky(*fp_except_t sticky*);

DESCRIPTION

The routines described herein are deprecated. New code should use the functionality provided by fenv(3).

When a floating point exception is detected, the exception sticky flag is set and the exception mask is tested. If the mask is set, then a trap occurs. These routines allow both setting the floating point exception masks, and resetting the exception sticky flags after an exception is detected. In addition, they allow setting the floating point rounding mode and precision.

The **fpgetround**() function returns the current floating point rounding mode.

The **fpsetround**() function sets the floating point rounding mode and returns the previous mode.

The **fpgetprec**() function returns the current floating point precision.

The **fpsetprec**() function sets the floating point precision and returns the previous precision.

The **fpgetmask**() function returns the current floating point exception masks.

The **fpsetmask**() function sets the floating point exception masks and returns the previous masks.

The **fpgetsticky**() function returns the current floating point sticky flags.

The **fpresetsticky**() function clears the floating point sticky flags and returns the previous flags.

Sample code which prevents a trap on divide-by-zero:

```
fpsetmask(~FP_X_DZ);
a = 1.0;
```

```
b = 0;
c = a / b;
fpresetsticky(FP_X_DZ);
fpsetmask(FP_X_DZ);
```

IMPLEMENTATION NOTES

The **fpgetprec**() and **fpsetprec**() functions provide functionality unavailable on many platforms. At present, they are implemented only on the i386 and amd64 platforms. Changing precision is not a supported feature: it may be ineffective when code is compiled to take advantage of SSE, and many library functions and compiler optimizations depend upon the default precision for correct behavior.

SEE ALSO

fenv(3), isnan(3)

HISTORY

These routines are based on SysV/386 routines of the same name.

CAVEATS

After a floating point exception and before a mask is set, the sticky flags must be reset. If another exception occurs before the sticky flags are reset, then a wrong exception type may be signaled.