fgetwc
getwc
getwchar
fgetws

There's another twist that affects the wide-character input functions. A call of a function that reads a single character (`fgetwc`, `getwc`, and `getwchar`) may fail because the bytes found in the input stream don't form a valid wide character or there aren't enough bytes available. The result is an encoding error, which causes the function to store `EILSEQ` in `errno` and return `WEOF`. The `fgetws` function, which reads a string of wide characters, may also fail because of an encoding error, in which case it returns a null pointer.

fputwc
putwc
putwchar
fputws

Wide-character output functions may also encounter encoding errors. Functions that write a single character (`fputwc`, `putwc`, and `putwchar`) store `EILSEQ` in `errno` and return `WEOF` if an encoding error occurs. However, the `fputws` function, which writes a wide-character string, is different: it returns `EOF` (not `WEOF`) if an encoding error occurs.

fwide

The `fwide` function doesn't correspond to any C89 function. `fwide` is used to determine the current orientation of a stream and, if desired, attempt to set its orientation. The `mode` parameter determines the behavior of the function:

- `mode` > 0. Attempts to make the stream wide-oriented if it has no orientation.
- `mode` < 0. Attempts to make the stream byte-oriented if it has no orientation.
- `mode` = 0. The orientation is not changed.

`fwide` doesn't change the orientation if the stream already has one.

The value returned by `fwide` depends on the orientation of the stream *after* the call. The return value is positive if the stream has wide orientation, negative if it has byte orientation, and zero if it has no orientation.

General Wide-String Utilities

The `<wchar.h>` header provides a number of functions that perform operations on wide strings. These are wide-character versions of functions that belong to the `<stdlib.h>` and `<string.h>` headers.

Wide-String Numeric Conversion Functions

```
double wcstod(const wchar_t * restrict nptr,
              wchar_t ** restrict endptr);
float wcstof(const wchar_t * restrict nptr,
             wchar_t ** restrict endptr);
long double wcstold(const wchar_t * restrict nptr,
                    wchar_t ** restrict endptr);
long int wcstol(const wchar_t * restrict nptr,
                wchar_t ** restrict endptr,
                int base);
long long int wcstoll(const wchar_t * restrict nptr,
                      wchar_t ** restrict endptr,
                      int base);
```