NAME

fseek, rewind, ftell – reposition a file pointer in a stream

SYNOPSIS

#include <stdio.h>

int fseek(FILE **stream***, long** *offset***, int** *ptrname***);**

void rewind(FILE **stream***);**

long ftell(FILE **stream***);**

MT-LEVEL

MT-Safe

DESCRIPTION

fseek( ) sets the position of the next input or output operation on the *stream* (see **intro**(3)). The new position is at the signed distance *offset* bytes from the beginning, from the current position, or from the end of the file, according to a *ptrname* value of **SEEK_SET**, **SEEK_CUR**, or **SEEK_END** (defined in **<stdio.h>**) as follows:

SEEK_SET set position equal to *offset* bytes.

SEEK_CUR set position to current location plus *offset*.

SEEK_END set position to **EOF** plus *offset*.

fseek( ) allows the file position indicator to be set beyond the end of the existing data in the file. If data is later written at this point, subsequent reads of data in the gap will return zero until data is actually written into the gap. **fseek( )**, by itself, does not extend the size of the file.

rewind(stream) is equivalent to:

(void) fseek (stream, 0L, SEEK_SET);

except that **rewind( )** also clears the error indicator on *stream*.

fseek( ) and **rewind( )** clear the **EOF** indicator and undo any effects of **ungetc( )** on *stream*. After **fseek( )** or **rewind( )**, the next operation on a file opened for update may be either input or output.

If *stream* is writable and buffered data has not been written to the underlying file, **fseek( )** and **rewind( )** cause the unwritten data to be written to the file.

ftell( ) returns the offset of the current byte relative to the beginning of the file associated with the named *stream*.

RETURN VALUES

fseek( ) returns –1 for improper seeks, otherwise zero. An improper seek can be, for example, an **fseek( )** done on a file that has not been opened via **fopen( )**; in particular, **fseek( )** may not be used on a terminal or on a file opened via **popen( )**. After a stream is closed, no further operations are defined on that stream.

SEE ALSO

lseek(2), **write**(2), **intro**(3), **fopen**(3S), **popen**(3S), **stdio**(3S), **ungetc**(3S)

NOTES

Although on the UNIX system an offset returned by **ftell( )** is measured in bytes, and it is permissible to seek to positions relative to that offset, portability to non-UNIX systems requires that an offset be used by **fseek( )** directly. Arithmetic may not meaningfully be performed on such an offset, which is not necessarily measured in bytes.