ftell

Defined in header <stdio.h>

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

Returns the file position indicator for the file stream stream.

If the stream is open in binary mode, the value obtained by this function is the number of bytes from the beginning of the file.

If the stream is open in text mode, the value returned by this function is unspecified and is only meaningful as the input to fseek().

Parameters

stream - file stream to examine

Return value

File position indicator on success or `-1L` if failure occurs.

On error, the `errno` variable is set to implementation-defined positive value.

Example

Demonstrates ftell() with error checking. Writes then reads a few floating-point (FP) values to/from a file.

Run this code

```
#include <stdio.h>
#include <stdlib.h>

/* If the condition is not met then exit the program with error message. */
void check(_Bool condition, const char* func, int line)
{
    if (condition)
        return;
    perror(func);
    fprintf(stderr, "%s failed in file %s at line # %d\n", func, __FILE__, line - 1);
    exit(EXIT_FAILURE);
}

int main(void)
{
    /* Prepare an array of FP values. */
    #define SIZE 5
    double A[SIZE] = {1.1,2.,3.,4.,5.};

    /* Write array to a file. */
    const char* fname = "/tmp/test.bin";
    FILE* file = fopen(fname, "wb");
    check(file != NULL, "fopen()", __LINE__);

    const int write_count = fwrite(A, sizeof(double), SIZE, file);
    check(write_count == SIZE, "fwrite()", __LINE__);

    fclose(file);

    /* Read the FP values into array B. */
    double B[SIZE];
    file = fopen(fname, "rb");
    check(file != NULL, "fopen()", __LINE__);

    long int pos = ftell(file); /* position indicator at start of file */
    check(pos != -1L, "ftell()", __LINE__);
    printf("pos: %ld\n", pos);

    const int read_count = fread(B, sizeof(double), 1, file); /* read one FP value */
    check(read_count == 1, "fread()", __LINE__);
```

```
    pos = ftell(file); /* position indicator after reading one FP value */
    check(pos != -1L, "ftell()", __LINE__);
    printf("pos: %ld\n", pos);
    printf("B[0]: %.1f\n", B[0]); /* print one FP value */

    return EXIT_SUCCESS;
}
```

Possible output:

```
pos: 0
pos: 8
B[0]: 1.1
```

References

- C11 standard (ISO/IEC 9899:2011):
  - 7.21.9.4 The ftell function (p: 337–338)
- C99 standard (ISO/IEC 9899:1999):
  - 7.19.9.4 The ftell function (p: 303–304)
- C89/C90 standard (ISO/IEC 9899:1990):
  - 4.9.9.4 The ftell function

See also

fgetpos	gets the file position indicator (function)
fseek	moves the file position indicator to a specific location in a file (function)
fsetpos	moves the file position indicator to a specific location in a file (function)

C++ documentation for **ftell**