Standard library header <cstdio>

This header was originally in the C standard library as <stdio.h>.

This header is part of the C-style input/output library.

Types

FILE	object type, capable of holding all information needed to control a C I/O stream (typedef)
fpos_t	complete non-array object type, capable of uniquely specifying a position in a file, including its multibyte parse state (typedef)
size_t	unsigned integer type returned by the sizeof operator (typedef)

Macros

NULL	implementation-defined null pointer constant (macro constant)
stdin **stdout** **stderr**	expression of type FILE* associated with the input stream expression of type FILE* associated with the output stream expression of type FILE* associated with the error output stream (macro constant)
EOF	integer constant expression of type int and negative value (macro constant)
FOPEN_MAX	number of files that can be open simultaneously (macro constant)
FILENAME_MAX	size needed for an array of char to hold the longest supported file name (macro constant)
BUFSIZ	size of the buffer used by std::setbuf (macro constant)
_IOFBF **_IOLBF** **_IONBF**	argument to std::setbuf indicating fully buffered I/O argument to std::setbuf indicating line buffered I/O argument to std::setbuf indicating unbuffered I/O (macro constant)
SEEK_SET **SEEK_CUR** **SEEK_END**	argument to std::fseek indicating seeking from beginning of the file argument to std::fseek indicating seeking from the current file position argument to std::fseek indicating seeking from end of the file (macro constant)
TMP_MAX	maximum number of unique filenames that can be generated by std::tmpnam (macro constant)
L_tmpnam	size needed for an array of char to hold the result of std::tmpnam (macro constant)

Functions

File access

fopen	opens a file (function)
freopen	open an existing stream with a different name (function)
fclose	closes a file (function)
fflush	synchronizes an output stream with the actual file (function)
setbuf	sets the buffer for a file stream (function)
setvbuf	sets the buffer and its size for a file stream (function)

Direct input/output

fread	reads from a file (function)

`fwrite`	writes to a file (function)

Unformatted input/output

Narrow character

`fgetc` `getc`	gets a character from a file stream (function)
`fgets`	gets a character string from a file stream (function)
`fputc` `putc`	writes a character to a file stream (function)
`fputs`	writes a character string to a file stream (function)
`getchar`	reads a character from `stdin` (function)
`gets` (deprecated in C++11) (removed in C++14)	reads a character string from `stdin` (function)
`putchar`	writes a character to `stdout` (function)
`puts`	writes a character string to `stdout` (function)
`ungetc`	puts a character back into a file stream (function)

Formatted input/output

Narrow/multibyte character

`scanf`
`fscanf`
`sscanf`
	reads formatted input from `stdin`, a file stream or a buffer
	(function)

`vscanf` (C++11) `vfscanf` (C++11) `vsscanf` (C++11)	reads formatted input from `stdin`, a file stream or a buffer using variable argument list (function)

`printf`
`fprintf`
`sprintf`
`snprintf` (C++11)
	prints formatted output to `stdout`, a file stream or a buffer
	(function)

`vprintf`
`vfprintf`
`vsprintf`
`vsnprintf` (C++11)
	prints formatted output to `stdout`, a file stream or a buffer using variable argument list
	(function)

File positioning

`ftell`	returns the current file position indicator (function)
`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)
`rewind`	moves the file position indicator to the beginning in a file (function)

Error handling

`clearerr`	clears errors (function)
`feof`	checks for the end-of-file (function)
`ferror`	checks for a file error (function)
`perror`	displays a character string corresponding of the current error to `stderr` (function)

Operations on files

`remove`	erases a file (function)
`rename`	renames a file (function)
`tmpfile`	creates and opens a temporary, auto-removing file (function)
`tmpnam`	returns a unique filename (function)

Synopsis

```
namespace std {
  using size_t = /* see description */;
  using FILE = /* see description */;
  using fpos_t = /* see description */;
}

#define NULL /* see description */
#define _IOFBF /* see description */
#define _IOLBF /* see description */
#define _IONBF /* see description */
#define BUFSIZ /* see description */
#define EOF /* see description */
#define FOPEN_MAX /* see description */
#define FILENAME_MAX /* see description */
#define L_tmpnam /* see description */
#define SEEK_CUR /* see description */
#define SEEK_END /* see description */
#define SEEK_SET /* see description */
#define TMP_MAX /* see description */
#define stderr /* see description */
#define stdin /* see description */
#define stdout /* see description */

namespace std {
  int remove(const char* filename);
  int rename(const char* old_p, const char* new_p);
  FILE* tmpfile();
  char* tmpnam(char* s);
  int fclose(FILE* stream);
  int fflush(FILE* stream);
  FILE* fopen(const char* filename, const char* mode);
  FILE* freopen(const char* filename, const char* mode, FILE* stream);
  void setbuf(FILE* stream, char* buf);
  int setvbuf(FILE* stream, char* buf, int mode, size_t size);
  int fprintf(FILE* stream, const char* format, ...);
  int fscanf(FILE* stream, const char* format, ...);
  int printf(const char* format, ...);
  int scanf(const char* format, ...);
  int snprintf(char* s, size_t n, const char* format, ...);
  int sprintf(char* s, const char* format, ...);
  int sscanf(const char* s, const char* format, ...);
  int vfprintf(FILE* stream, const char* format, va_list arg);
  int vfscanf(FILE* stream, const char* format, va_list arg);
  int vprintf(const char* format, va_list arg);
  int vscanf(const char* format, va_list arg);
  int vsnprintf(char* s, size_t n, const char* format, va_list arg);
  int vsprintf(char* s, const char* format, va_list arg);
  int vsscanf(const char* s, const char* format, va_list arg);
  int fgetc(FILE* stream);
  char* fgets(char* s, int n, FILE* stream);
  int fputc(int c, FILE* stream);
  int fputs(const char* s, FILE* stream);
  int getc(FILE* stream);
  int getchar();
  int putc(int c, FILE* stream);
  int putchar(int c);
  int puts(const char* s);
  int ungetc(int c, FILE* stream);
  size_t fread(void* ptr, size_t size, size_t nmemb, FILE* stream);
  size_t fwrite(const void* ptr, size_t size, size_t nmemb, FILE* stream);
  int fgetpos(FILE* stream, fpos_t* pos);
  int fseek(FILE* stream, long int offset, int whence);
  int fsetpos(FILE* stream, const fpos_t* pos);
  long int ftell(FILE* stream);
  void rewind(FILE* stream);
  void clearerr(FILE* stream);
  int feof(FILE* stream);
  int ferror(FILE* stream);
  void perror(const char* s);
}
```

Notes

- NULL is also defined in the following headers:
  - <clocale>
  - <ctime>
  - <cstddef>
  - <cstring>
  - <cwchar>
  - <cstdlib>
- std::size_t is also defined in the following headers:
  - <ctime>
  - <cstddef>
  - <cstring>
  - <cwchar>
  - <cuchar> (since C++17)
  - <cstdlib>

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/header/cstdio&oldid=120995"