File input/output

The <stdio.h> header provides generic file operation support and supplies functions with narrow character input/output capabilities.

The <wchar.h> header supplies functions with wide character input/output capabilities.

I/O streams are denoted by objects of type FILE that can only be accessed and manipulated through pointers of type `FILE*`. Each stream is associated with an external physical device (file, standard input stream, printer, serial port, etc).

Types

Defined in header <stdio.h>

FILE	object type, capable of holding all information needed to control a C I/O stream (typedef)
fpos_t	non-array complete object type, capable of uniquely specifying a position and multibyte parser state in a file (typedef)

Predefined standard streams

Defined in header <stdio.h>

stdin	expression of type `FILE*` associated with the input stream
stdout	expression of type `FILE*` associated with the output stream
stderr	expression of type `FILE*` associated with the error output stream (macro constant)

Functions

File access

Defined in header <stdio.h>

fopen **fopen_s** (C11)	opens a file (function)
freopen **freopen_s** (C11)	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)

Defined in header <wchar.h>

fwide (C95)	switches a file stream between wide character I/O and narrow character I/O (function)

Direct input/output

Defined in header <stdio.h>

fread	reads from a file (function)
fwrite	writes to a file (function)

Unformatted input/output

Narrow character

Defined in header <stdio.h>

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 (removed in C11) **gets_s** (C11)	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)

Wide character

Defined in header <wchar.h>

fgetwc (C95) **getwc**	gets a wide character from a file stream (function)
fgetws (C95)	gets a wide string from a file stream (function)
fputwc (C95) **putwc**	writes a wide character to a file stream (function)
fputws (C95)	writes a wide string to a file stream (function)
getwchar (C95)	reads a wide character from stdin (function)
putwchar (C95)	writes a wide character to stdout (function)
ungetwc (C95)	puts a wide character back into a file stream (function)

Formatted input/output

Narrow character

Defined in header <stdio.h>

scanf **fscanf** **sscanf** **scanf_s** (C11) **fscanf_s** (C11) **sscanf_s** (C11)	reads formatted input from stdin, a file stream or a buffer (function)
vscanf (C99) **vfscanf** (C99) **vsscanf** (C99) **vscanf_s** (C11) **vfscanf_s** (C11) **vsscanf_s** (C11)	reads formatted input from stdin, a file stream or a buffer using variable argument list (function)
printf **fprintf** **sprintf** **snprintf** (C99) **printf_s** (C11) **fprintf_s** (C11) **sprintf_s** (C11) **snprintf_s** (C11)	prints formatted output to stdout, a file stream or a buffer (function)
vprintf **vfprintf** **vsprintf** **vsnprintf** (C99) **vprintf_s** (C11) **vfprintf_s** (C11) **vsprintf_s** (C11) **vsnprintf_s** (C11)	prints formatted output to stdout, a file stream or a buffer using variable argument list (function)

Wide character

Defined in header <wchar.h>

	reads formatted wide character input from stdin, a file stream or a buffer

wscanf (C95) **fwscanf** (C95) **swscanf** (C95) **wscanf_s** (C11) **fwscanf_s** (C11) **swscanf_s** (C11)	(function)
vwscanf (C99) **vfwscanf** (C99) **vswscanf** (C99) **vwscanf_s** (C11) **vfwscanf_s** (C11) **vswscanf_s** (C11)	reads formatted wide character input from stdin, a file stream or a buffer using variable argument list (function)
wprintf (C95) **fwprintf** (C95) **swprintf** (C95) **wprintf_s** (C11) **fwprintf_s** (C11) **swprintf_s** (C11) **snwprintf_s** (C11)	prints formatted wide character output to stdout, a file stream or a buffer (function)
vwprintf (C95) **vfwprintf** (C95) **vswprintf** (C95) **vwprintf_s** (C11) **vfwprintf_s** (C11) **vswprintf_s** (C11) **vsnwprintf_s** (C11)	prints formatted wide character output to stdout, a file stream or a buffer using variable argument list (function)

File positioning

Defined in header <stdio.h>

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

Defined in header <stdio.h>

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

Defined in header <stdio.h>

remove	erases a file (function)
rename	renames a file (function)
tmpfile **tmpfile_s** (C11)	returns a pointer to a temporary file (function)
tmpnam **tmpnam_s** (C11)	returns a unique filename (function)

Macro constants

Defined in header <stdio.h>

EOF	integer constant expression of type `int` and negative value (macro constant)
FOPEN_MAX	maximum 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 setbuf() (macro constant)
_IOFBF **_IOLBF** **_IONBF**	argument to setvbuf() indicating fully buffered I/O argument to setvbuf() indicating line buffered I/O argument to setvbuf() indicating unbuffered I/O (macro constant)
SEEK_SET **SEEK_CUR** **SEEK_END**	argument to fseek() indicating seeking from beginning of the file argument to fseek() indicating seeking from the current file position argument to fseek() indicating seeking from end of the file (macro constant)
TMP_MAX **TMP_MAX_S** (C11)	maximum number of unique filenames that can be generated by tmpnam maximum number of unique filenames that can be generated by tmpnam_s (macro constant)
L_tmpnam **L_tmpnam_s** (C11)	size needed for an array of `char` to hold the result of tmpnam size needed for an array of `char` to hold the result of tmpnam_s (macro constant)

References

- C11 standard (ISO/IEC 9899:2011):
  - 7.21 Input/output <stdio.h> (p: 296–339)
  - 7.29 Extended multibyte and wide character utilities <wchar.h> (p: 402–446)
  - 7.31.11 Input/output <stdio.h> (p: 456)
  - 7.31.16 Extended multibyte and wide character utilities <wchar.h> (p: 456)
  - K.3.5 Input/output <stdio.h> (p: 586–603)
- C99 standard (ISO/IEC 9899:1999):
  - 7.19 Input/output <stdio.h> (p: 262–305)
  - 7.24 Extended multibyte and wide character utilities <wchar.h> (p: 348–392)
  - 7.26.9 Input/output <stdio.h> (p: 402)
  - 7.26.12 Extended multibyte and wide character utilities <wchar.h> (p: 402)
- C89/C90 standard (ISO/IEC 9899:1990):
  - 4.9 INPUT/OUTPUT <stdio.h>
  - 4.13.6 Input/output <stdio.h>

See also

C++ documentation for **C-style file input/output**