

文件系统库

文件系统库提供在文件系统与其组件，例如路径、常规文件与目录上进行操作的设施。

文件系统库原作为 `boost.filesystem`

(<http://www.boost.org/doc/libs/release/libs/filesystem/doc/index.htm>) 开发，出版为技术规范 ISO/IEC TS 18822:2015，并最终从 C++17 开始并入 ISO C++。现在 boost 实现可用的编译器和平台于多于 C++17 库。

若层级文件系统不能为实现所访问，或若它不提供必要的兼容性，则文件系统库设施可能不可用。若底层文件系统不支持，则一些特性可能不可用（例如 FAT 文件系统缺少符号链接并禁止多重硬链接）。这些情况下必须报告错误。

若对此库的函数的调用引入文件系统竞争，即多个线程、进程或计算机交错地访问并修改文件系统中的同一对象，则行为未定义。

库范畴定义

- **文件**：持有数据的文件系统对象，能被写入或读取，或二者皆可。文件拥有名称及属性，属性之一是文件类型：
 - **目录**：表现为目录条目的容器的文件，目录条目标识其他文件（其中一些可以是另外的嵌套的目录）。讨论到具体文件时，包含该文件目录条目的目录是其父**目录**。父目录能以相对路径名 `".."` 表示。
 - **硬链接**：关联一个名字到一个既存文件的目录条目。若支持多重硬链接，则文件在最后一个到它的硬链接被移除后才被移除。
 - **符号链接**：关联一个名词到一个路径的目录条目，路径可以存在亦可不存在。
 - **常规文件**：不是其他文件类型的文件。
- **文件名**：命名一个文件的字符串。容许字符、大小写区别、最大长度以及被禁止名称是实现定义的。名称 `"."`（点）与 `".."`（双点）在库层次拥有特殊含义。
- **路径**：标识一个文件的元素序列。它以可选的 **根名**（例如 Windows 上的 `"C:"` 或 `"//server"`）开始，后随可选的 **根目录**（例如 Unix 上的 `"/"`），后随零或更多个文件名（除了最后一个都必须是目录或到目录的链接）的序列。表示路径的字符串（**路径名**）的原生格式（如哪些字符被用作分隔符）与字符编码是实现定义的，库提供路径的可移植表示。
 - **绝对路径**：无歧义地标识一个文件位置的路径。
 - **规范路径**：不包含符号链接、`"."` 或 `".."` 元素的绝对路径。
 - **相对路径**：标识相对于文件系统中某位置的文件位置的路径。特殊路径名 `"."`（点，“当前目录”）和 `".."`（双点，“父目录”）是相对路径。

类

在标头 `<filesystem>` 定义
在命名空间 `std::filesystem` 定义

path (C++17)	表示路径 (类)
filesystem_error (C++17)	文件系统错误时抛出的异常 (类)
directory_entry (C++17)	目录条目 (类)
directory_iterator (C++17)	指向目录内容的迭代器 (类)
recursive_directory_iterator (C++17)	指向一个目录及其子目录的内容的迭代器 (类)
file_status (C++17)	表示文件类型及权限 (类)
space_info (C++17)	关于文件系统上空闲及可用空间的信息 (类)
file_type (C++17)	文件的类型 (枚举)
perms (C++17)	标识文件系统权限 (枚举)
perm_options (C++17)	指定权限操作的语义 (枚举)
copy_options (C++17)	指定复制操作的语义 (枚举)
directory_options (C++17)	用于迭代目录内容的选项 (枚举)
file_time_type (C++17)	表示文件时间值 (typedef)

非成员函数

在标头 `<filesystem>` 定义
在命名空间 `std::filesystem` 定义

absolute (C++17)	组成一个绝对路径 (函数)
canonical weakly_canonical (C++17)	组成一个规范路径 (函数)
relative proximate (C++17)	组成一个相对路径 (函数)
copy (C++17)	复制文件或目录 (函数)
copy_file (C++17)	复制文件内容 (函数)
copy_symlink (C++17)	复制一个符号链接 (函数)
create_directory (C++17) create_directories (C++17)	创建新目录 (函数)
create_hard_link (C++17)	创建一个硬链接 (函数)

create_symlink (C++17) create_directory_symlink (C++17)	创建一个符号链接 (函数)
current_path (C++17)	返回或设置当前工作目录 (函数)
exists (C++17)	检查路径是否指代既存的文件系统对象 (函数)
equivalent (C++17)	检查两个路径是否指代同一文件系统对象 (函数)
file_size (C++17)	返回文件的大小 (函数)
hard_link_count (C++17)	返回指代特定文件的硬链接数 (函数)
last_write_time (C++17)	获取或设置最近一次数据修改的时间 (函数)
permissions (C++17)	修改文件访问权限 (函数)
read_symlink (C++17)	获得符号链接的目标 (函数)
remove (C++17) remove_all (C++17)	移除一个文件或空目录 移除一个文件或递归地移除一个目录及其所有内容 (函数)
rename (C++17)	移动或重命名一个文件或目录 (函数)
resize_file (C++17)	以截断或填充零更改一个常规文件的大小 (函数)
space (C++17)	确定文件系统上的可用空闲空间 (函数)
status (C++17) symlink_status (C++17)	确定文件属性 确定文件属性，检查符号链接目标 (函数)
temp_directory_path (C++17)	返回一个适用于临时文件的目录 (函数)

文件类型

is_block_file (C++17)	检查给定的路径是否表示块设备 (函数)
is_character_file (C++17)	检查给定的路径是否表示字符设备 (函数)
is_directory (C++17)	检查给定的路径是否表示一个目录 (函数)
is_empty (C++17)	检查给定的路径是否表示一个空文件或空目录 (函数)
is_fifo (C++17)	检查给定的路径是否表示一个命名管道 (函数)
is_other (C++17)	检查参数是否表示一个其他文件 (函数)
is_regular_file (C++17)	检查参数是否表示一个常规文件 (函数)

is_socket (C++17)	检查参数是否表示一个具名 IPC 套接字 (函数)
is_symlink (C++17)	检查参数是否表示一个符号链接 (函数)
status_known (C++17)	检查文件状态是否已知 (函数)

注解

使用此库可能要求额外的编译器/链接器选项。9.1 前的 GNU 实现要求用 `-lstdc++fs` 链接，而 LLVM 9.0 前的 LLVM 实现要求用 `-lc++fs` 链接。

参阅

文件系统 **TS** 的 C++ 文档

来自“<https://zh.cppreference.com/mwiki/index.php?title=cpp/filesystem&oldid=73286>”