标准 C 语言头文件

ISO C 标准定义的头文件(24 项)

<assert.h> 验证程序断言　<complex.h> 支持复数算术运算　<ctype.h> 字符类型

<errno.h> 出错码　<fenv.h> 浮点环境　<float.h> 浮点常量

<inttypes.h> 整型格式转换　<iso646.h> 替代关系操作符宏　<limits.h> 实现常量

<locale.h> 局部类别　<math.h> 数学常量　<setjmp.h> 非局部 goto

<signal.h> 信号　<stdarg.h> 可变参数表　<stdbool.h> 布尔类型和值

<stddef.h> 标准定义　<stdint.h> 整型　<stdio.h> 标准 I/O 库

<stdlib.h> 实用程序库函数　<string.h> 字符串操作　<tgmath.h> 通用类型数学宏

<time.h> 时间和日期　<wchar.h> 宽字符支持　<wctype.h> 宽字符分类和映射支持

POSIX 标准定义的必须的头文件(26 项)

<dirent.h> 目录项　<fcntl.h> 文件控制　<fnmatch.h> 文件名匹配类型

<glob.h> 路径名模式匹配类型　<grp.h> 组文件　<netdb.h> 网络数据库操作

<pwd.h> 口令文件　<regex.h> 正则表达式　<tar.h> tar 归档值

<termios.h> 终端 I/O　<unistd.h> 符号常量　<utime.h> 文件时间

<wordexp.h> 字扩展类型　<arpa/inet.h> Internet 定义　<net/if.h> 套接字本地接口

<netinet/in.h> Internet 地址族　<netinet/tcp.h> 传输控制协议　<sys/mman.h> 内存管理声明

<sys/select.h> select 函数　<sys/socket.h> 套接字接口　<sys/stat.h> 文件状态

<sys/times.h> 进程时间　<sys/types.h> 基本系统数据类型　<sys/un.h> UNIX 域套接字定义

<sys/utsname.h>系统名　<sys/wait.h> 进程控制

POSIX 标准定义的 XSI 扩展头文件(26 项)

<cpio.h> cpio 归档值　<dlfcn.h> 动态链接　<fmtmsg.h> 消息显示结构

<ftw.h> 文件树漫游　<iconv.h> 代码集转换实用程序　<langinfo.h> 语言信息常量

<libgen.h> 模式匹配函数定义　<monetary.h> 货币类型　<ndbm.h> 数据库操作

<nl_types.h> 消息类别　<poll.h> 轮询函数　<search.h> 搜索表

<strings.h> 字符串操作　<syslog.h> 系统出错日志记录　<ucontext.h> 用户上下文

<ulimit.h> 用户限制　<utmpx.h> 用户帐户数据库　<sys/ipc.h> IPC

<sys/msg.h> 消息队列　<sys/resource.h> 资源操作　<sys/sem.h> 信号量

<sys/shm.h> 共享存储　<sys/statvfs.h> 文件系统信息　<sys/time.h> 时间类型

<sys/timeb.h> 附加的时间　<sys/uio.h> 矢量 I/O 操作

POSIX 标准定义的可选头文件(8 项)

<aio.h> 异步 I/O　<mqueue.h> 消息队列　<pthread.h> 线程

<sched.h> 执行调度　<semaphore.h> 信号量　<spawn.h> 实时 spawn 接口

<stropts.h> XSI STREAMS 接口　<trace.h> 时间跟踪

标准 C++ 语言头文件（54 个其中 16 个用于构建 STL,3 个为附加非必须）

<algorithm> STL 通用算法	<bitset> STL 位集容器	<cassert> 用于在程序运行时执行断言
<cctype> 字符处理	<cerrno> 错误码	<cfloat> 用于测试浮点类型属性
<ciso646> ISO646 变体字符集	<climits> 测试整数类型属性	<clocale> 本地化函数
<cmath> 数学函数	<complex> 复数类	<csetjmp> 执行非内部的 goto 语句
<csignal> 信号	<cstdarg> 访问参数数量变化的函数	<cstddef> 用于定义实用的类型和宏
<cstdio> 输入/输出	<cstdlib> 杂项函数及内存分配	<cstring> 字符串
<ctime> 时间	<cwchar> 宽字符处理及输入/输出	<cwctype> 宽字符分类
<deque> STL 双端队列容器	<exception> 异常处理类	<fstream> 文件流
<functional> STL 函数对象	<iomanip> 参数化输入/输出	<ios> 基本输入/输出支持
<iosfwd> 输入/输出前置声明	<iostream> 数据流输入/输出	<istream> 基本输入流
<iterator> 遍历序列的类	<limits> 各种数据类型最值常量	<list> STL 线性列表容器
<locale> 国际化支持	<map> STL 映射容器	<memory> 专用内存分配器
<new> 基本内存分配和释放	<numeric> 通用的数字操作	<ostream> 基本输出流
<queue> STL 队列容器	<set> STL 集合容器	<sstream> 基于字符串的流
<stack> STL 堆栈容器	<stdexcept> 标准异常类	<streambuf> iostream 的缓冲区类
<string> 字符串类	<strstream> 非内存字符序列的流类	<typeinfo> 运行时类型标识
<utility> STL 通用模板类	<valarray> 支持值数组的类和模版类	<vector> STL 动态数组容器

标准 C++附加的头文件(3 个)非必须

<hash_map>	<hash_set>	<slist>

The Standard C++ library consists of 51 required headers.This implementation also includes three additional headers,<hash_map>,<hash_set>,and <slist>,not required by the C++ Standard, for a total of 54 headers.Of these 54 headers,16 constitute the Standard Template Library, or STL.These are indicated below with the notation

<algorithm> -- (STL) for defining numerous templates that implement useful algorithms
<bitset> -- for defining a template class that administers sets of bits
<complex> -- for defining a template class that supports complex arithmetic
<deque> -- (STL) for defining a template class that implements a deque container
<exception> -- for defining several functions that control exception handling
<fstream> -- for defining several iostreams template classes that manipulate exteral files
<functional> -- (STL) for defining several templates that help construct predicates for the templates defined in <algorithm> and <numeric>
<hash_map> -- (STL) for defining template classes that implement hashed associative containers that map keys to values
<hash_set> -- (STL) for defining template classes that implement hashed associative containers
<iomanip> -- for declaring several iostreams manipulators that take an argument
<ios> -- for defining the template class that serves as the base for many iostreams classes
<iosfwd> -- for declaring several iostreams template classes before they are necessarily defined
<iostream> -- for declaring the iostreams objects that manipulate the standard streams
<istream> -- for defining the template class that performs extractions
<iterator> -- (STL) for defining several templates that help define and manipulate iterators
<limits> -- for testing numeric type properties
<list> -- (STL) for defining a template class that implements a doubly linked list container
<locale> -- for defining several classes and templates that control locale-specific behavior, as in the iostreams classes
<map> -- (STL) for defining template classes that implement associative containers that map keys to values
<memory> -- (STL) for defining several templates that allocate and free storage for various container classes
<new> -- for declaring several functions that allocate and free storage
<numeric> -- (STL) for defining several templates that implement useful numeric functions
<ostream> -- for defining the template class that performs insertions
<queue> -- (STL) for defining a template class that implements a queue container
<set> -- (STL) for defining template classes that implement associative containers
<slist> -- (STL) for defining a template class that implements a singly linked list container
<sstream> -- for defining several iostreams template classes that manipulate string containers
<stack> -- (STL) for defining a template class that implements a stack container
<stdexcept> -- for defining several classes useful for reporting exceptions
<streambuf> -- for defining template classes that buffer iostreams operations
<string> -- for defining a template class that implements a string container
<strstream> -- for defining several iostreams classes that manipulate in-memory character sequences
<typeinfo> -- for defining class type_info, the result of the typeid operator
<utility> -- (STL) for defining several templates of general utility
<valarray> -- for defining several classes and template classes that support value-oriented arrays

<vector> -- (STL) for defining a template class that implements a vector container

新的 C 标准库

<cassert> -- for enforcing assertions when functions execute
<cctype> -- for classifying characters
<cerrno> -- for testing error codes reported by library functions
<cfloat> -- for testing floating-point type properties
<ciso646> -- for programming in ISO 646 variant character sets
<climits> -- for testing integer type properties
<clocale> -- for adapting to different cultural conventions
<cmath> -- for computing common mathematical functions
<csetjmp> -- for executing nonlocal goto statements
<csignal> -- for controlling various exceptional conditions
<cstdarg> -- for accessing a varying number of arguments
<cstddef> -- for defining several useful types and macros
<cstdio> -- for performing input and output
<cstdlib> -- for performing a variety of operations
<cstring> -- for manipulating several kinds of strings
<ctime> -- for converting between various time and date formats
<cwchar> -- for manipulating wide streams and several kinds of strings
<cwctype> -- for classifying wide characters

旧的 C 标准库

<assert.h> -- for enforcing assertions when functions execute
<ctype.h> -- for classifying characters
<errno.h> -- for testing error codes reported by library functions
<float.h> -- for testing floating-point type properties
<iso646.h> -- for programming in ISO 646 variant character sets
<limits.h> -- for testing integer type properties
<locale.h> -- for adapting to different cultural conventions
<math.h> -- for computing common mathematical functions
<setjmp.h> -- for executing nonlocal goto statements
<signal.h> -- for controlling various exceptional conditions
<stdarg.h> -- for accessing a varying number of arguments
<stddef.h> -- for defining several useful types and macros
<stdio.h> -- for performing input and output
<stdlib.h> -- for performing a variety of operations
<string.h> -- for manipulating several kinds of strings
<time.h> -- for converting between various time and date formats
<wchar.h> -- for manipulating wide streams and several kinds of strings
<wctype.h> -- for classifying wide characters

Finally, in this implementation, the Standard C++ library also includes several headers for compatibility with traditional C++ libraries:

<fstream.h> -- for defining several iostreams template classes that manipulate exteral files
<iomanip.h> -- for declaring several iostreams manipulators that take an argument
<iostream.h> -- for declaring the iostreams objects that manipulate the standard streams
<new.h> -- for declaring several functions that allocate and free storage
<stl.h> -- for declaring several template classes that aid migration from older versions of the Standard Template Library