

Windows与Linux下的库介绍 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/462/2021_2022_Windows_E4_B8_8E_c103_462218.htm

什么是库 库文件是一些预先编译好的函数的集合，那些函数都是按照可再使用的原则编写的。它们通常由一组互相关联的用来完成某项常见工作的函数构成，从本质上来说库是一种可执行代码的二进制形式，可以被操作系统载入内存执行。无论在Windows平台还是在Linux平台下都存在大量的库，但由于Windows和Linux的本质不同，因此二者的库的二进制是不兼容的。库的分类 Windows下的库有两种：静态库（.lib）和动态链接库（.dll）。Linux下的库有两种：静态库（.a）和共享库（.so）。Linux下的静态库的名字一般为libxxx.a，其中xxx是该lib的名称Linux下的动态库的名字一般为libxxx.so.major.minor，xxx是该lib的名称，major是主版本号，minor是副版本号。

Windows下的库与Linux下的库的异同 Linux的共享库（.so）就象Windows的动态链接库（.dll），它里面包含有很多程序常用的函数。为了方便程序开发和减少程序的冗余，程序当中就不用包含每个常用函数的拷贝，只是在需要时调用共享库中常函数即可。这种方式我们称之为动态链接

（Dynamically Linked）。而有时我们不希望叫程序去调用共享库的函数，而是把库函数代码直接链接进程序代码中，也就是说，在程序本身拥有一份共享库中函数的副本。这种方式我们称之为静态链接（Statically Linked）。所以，简单的讲静态库和共享库（动态库）的不同点在于代码被载入的时刻不同。静态库的代码在编译过程中已经被载入可执行程序，

因此体积较大。共享库的代码是在可执行程序运行时才载入内存的，在编译过程中仅简单的引用，因此代码体积较小。Windows下的动态链接库（.dll）与Linux下的共享库（.so）的差别。dll文件事实上和.exe文件一样，同属PE格式的-executable文件。对于隐式的引用外部符号，需要把外部符号所在的位置写在PE头上。PE加载器将从PE头上找到依赖的符号表，并加载依赖的其它.dll文件。而在Linux上并非如此！.so文件大多为elf-executable文件格式。当它们需要的外部符号，可以不写明这些符号所在的位置。也就是说，通常.so文件本身并不知道它依赖的那些符号在哪些.so里面。这些符号是由调用dlopen的进程运行时提供的。我们在Windows下做一个.dll文件时还需要携带一个.lib文件；而在Linux下一般只需要有相应的头文件就够了。对于编写新的.so，找不到的符号可以就让它在那里，直到最终-executable文件来把所有需要的符号联合到一起。windows可以存在一个.dll对另一个.dll的隐式依赖；而Linux下一般不需要让.so和.so有隐式依赖关系。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com