**1、静态库（Static Library）：**

**·文件后缀：**静态库的文件通常以.a（在Windows上可能是.lib）为后缀，表示静态库。

**·链接方式：**在编译时，静态库的代码被复制并链接到可执行文件中，形成一个完整的可执行文件。

**·运行时：**静态库的代码在程序启动时就被载入内存，与程序的其他部分一起组成一个单独的可执行文件。因此，静态库的大小相对较大。

**·独立性：**程序运行时不依赖于外部的库文件，所有的代码都被打包到可执行文件中。

**·部署：**分发程序时，需要将所有依赖的库文件一同发布，程序在任何环境中都能独立运行。

**2、动态库（Dynamic Library）：**

**·文件后缀：** 动态库的文件通常以.so（在Windows上可能是.dll）为后缀，表示动态库。

**·链接方式：** 在编译时，程序仅包含对动态库的引用，而不包含实际的代码。链接过程在运行时进行。

**·运行时：** 动态库的代码在程序运行时才被载入内存，多个程序可以共享同一个动态库的实例，从而减小了内存占用。

**·独立性：** 程序在运行时仍依赖于外部的动态库文件，因此需要确保在运行环境中有相应的动态库。

**·部署：** 分发程序时，只需要携带动态库的引用，用户需要在其系统上安装相应版本的动态库。

**3、优缺点比较：**

**静态库优点：**

·在编译时就被链接，因此不受运行时库版本的影响。

·不需要在运行时解析和加载库，因此运行时性能可能更好。

**静态库缺点：**

·可执行文件较大，包含了库的全部代码。

·需要重新编译整个程序，才能使用库的新版本。

**动态库优点：**

·可以在不重新编译程序的情况下更新或替换库。

·多个程序可以共享同一份库的实例，减小内存占用。

**动态库缺点：**

·需要在运行时解析和加载库，可能引入一些性能开销。

·对于程序的用户来说，需要确保系统中有正确版本的动态库。