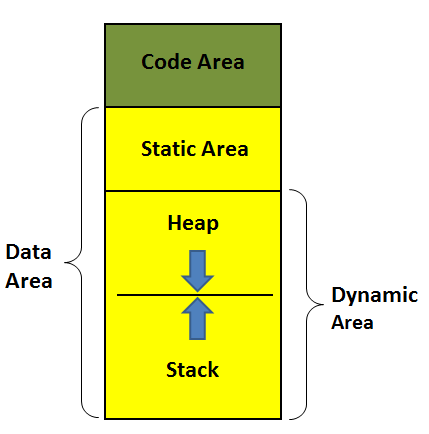
1. c++编译型语言，编译器编译到目标平台后就可以跨平台使用，java需要jre要解释到目标平台的机器语言实现跨平台，c#依赖于.net，但.net跨平台不行，mono可以跨平台，所以c#可以通过mono跨平台，jit即时编译到目标平台代码。
2. C Runtime是windows平台下的一个概念。一般来说，CRT函数就是标准的C语言函数。例如，printf、scC Runtime是windows平台下的一个概念。一般来说，CRT函数就是标准的C语言函数。例如，printf、scanf、strlen、fopen等函数就属于CRT函数。anf、strlen、fopen等函数就属于CRT函数。
3. .so是安卓系统的动态链接库，类似于windows系统的c++的dll，JNI（Java native interface）属于java，与c\c++互相调用的接口，NDK（native development kit）属于Android，用于在安卓上使用JNI。
4. ABI（Application Binary Interface，应用二进制接口），ABI定义了其所对应的CPU架构能够执行的二进制文件（如.so文件）的格式规范。
5. 编译型语言c\c++，解释型语言Java\c#（静态），python\lua（动态,也成为脚本语言）。
6. 程序和程序之间的内存是独立的，不能互相访问，比如QQ和浏览器分别所占的内存区域是不能相互访问的。而每个程序的内存也是分区管理的，一个应用程序所占的内存可以分为很多个区域，我们需要了解的主要有四个区域，通常叫内存四区:



**1.代码区**

　　程序被操作系统加载到内存的时候，所有的可执行代码（程序代码指令、常量字符串等）都加载到代码区，这块内存在程序运行期间是不变的。代码区是平行的，里面装的就是一堆指令，在程序运行期间是不能改变的。函数也是代码的一部分，故函数都被放在代码区，包括main函数。

　　注意："int a = 0;"语句可拆分成"int a;"和"a = 0"，定义变量a的"int a;"语句并不是代码，它在程序编译时就执行了，并没有放到代码区，放到代码区的只有"a = 0"这句。

**2.静态区**

　　静态区存放程序中所有的全局变量和静态变量。

**3.栈区**

　　栈（stack）是一种先进后出的内存结构，所有的自动变量、函数形参都存储在栈中，这个动作由编译器自动完成，我们写程序时不需要考虑。栈区在程序运行期间是可以随时修改的。当一个自动变量超出其作用域时，自动从栈中弹出。

* 每个线程都有自己专属的栈；
* 栈的最大尺寸固定，超出则引起栈溢出；
* 变量离开作用域后栈上的内存会自动释放。

## 4.堆区

　　堆（heap）和栈一样，也是一种在程序运行过程中可以随时修改的内存区域，但没有栈那样先进后出的顺序。更重要的是堆是一个大容器，它的容量要远远大于栈，这可以解决上面实验三造成的内存溢出困难。一般比较复杂的数据类型都是放在堆中。但是在C语言中，堆内存空间的申请和释放需要手动通过代码来完成。对于一个32位操作系统，最大管理管理4G内存，其中1G是给操作系统自己用的，剩下的3G都是给用户程序，一个用户程序理论上可以使用3G的内存空间。堆上的内存必须手动释放（C/C++），除非语言执行环境支持GC（如C#在.NET上运行就有垃圾回收机制）。对内存申请释放，malloc和free。