

C++内存管理

【深入理解】

代码段（code segment/text segment）：存放程序执行代码的一块内存区域。内存大小在程序运行前就已经确定，并且内存区域通常属于只读，某些架构也允许代码段为可写，即允许修改程序。在代码段中，也有可能包含一些只读的常数变量，例如字符串常量等。代码区的指令中包括操作码和要操作的对象（或对象地址引用）。如果是立即数（即具体的数值，如5），将直接包含在代码中；如果是局部数据，将在栈区分配空间，然后引用该数据地址；如果是BSS区和数据区，在代码中同样将引用该数据地址。另外，代码段还规划了局部数据所申请的内存空间信息。

数据段：data段中的静态数据区存放的是程序中已初始化的全局变量、静态变量和常量。

BSS 段（未初始化全局变量区）：BSS 段（bss segment）通常是指用来存放程序中未初始化的全局变量的一块内存区域。

栈(stack)：存放局部变量，不需要程序员显式地管理。

堆(heap)：用于动态内存分配，大小并不固定，可动态扩张或缩减。当进程调用malloc等函数分配内存时，新分配的内存就被动态添加到堆上（堆被扩张）；当利用free等函数释放内存时，被释放的内存从堆中被剔除（堆被缩减）。在将应用程序加载到内存空间执行时，操作系统负责代码段、数据段和BSS段的加载，并将在内存中为这些段分配空间。栈段亦由操作系统分配和管理，而不需要程序员显式地管理；堆段由程序员自己管理，即显式地申请和释放空间。

文字常量区：常量字符串就是放在这里，程序结束后有系统释放。

【内存的分配方式】

静态分配：编译器在处理程序源代码时分配。

动态分配：程序在执行时调用malloc库函数申请分配。

静态内存分配是在程序执行之前进行的因而效率比较高，而动态内存分配则可以灵活的处理未知数目的。

静态对象是有名字的变量，可以直接对其进行操作；动态对象是没有名字的一段地址，需要通过指针间接地对它进行操作。

静态对象的分配与释放由编译器自动处理；动态对象的分配与释放必须由程序员显式地管理，它通过malloc()和free两个函数来完成

【char *a 与 char a[]的区别】

char *a = "abcd"此时"abcd"存放在常量区。通过指针只可以访问字符串常量，而不可以改变它。而char a[20] = "abcd";此时"abcd"存放在栈。可以通过指针去访问和修改数组内容。

另外注意：char a[] = "01234"，虽然没有指明字符串的长度，但是此时系统已经开好了，就是大小为6----'0' '1' '2' '3' '4' '5' '\0'，(注意strlen(a)是不计'\0')