|  |
| --- |
|  |
| 栈内存与堆内存的区别 |
|  |
|  |
| |  | | --- | | 一:程序的内存分配      栈区（stack）有编译器自动分配释放，存放函数的参数，局部变量等.      堆区（heap）一般由程序员分配和释放，否则就由OS回收。堆用于存放全局变量，静态变量，常量字符串和函数代码（函数体的二进制代码).    二：申请后系统的响应         栈：只要栈的剩余空间大于所申请的空间，系统将为程序提供内存，否则将报异常提示栈溢出。         堆：在记录空闲内存地址的链表中寻找一个空间大于所申请空间的堆结点，然后将该结点从空闲结点链表中删除，并将该结点的空间分配给程序。另外，对于大多数系统会在这块内存空间的首地址出记录本次分配空间的大小，这样代码中的delete 才能正确释放本内存空间。系统会将多余的那部分重新空闲链表中。    三：申请大小的限制         栈：在windows中，栈是向低地址扩展的数据结构，是一块连续的内存区域。也就是栈顶的地址和栈的最大容量（windows是2M）是系统规定好的。如果申请的空间超过剩余栈空间，将提示overflow，因此从栈获得的空间较小。         堆：堆是向高地址扩展的数据结构，是不连续的内存区域。堆的大小受限于系统中有效的虚拟内存。由此可见堆的内存获得比较灵活，空间较大。    四：栈和堆的存储内容         栈：在栈中，第一个进栈的是主函数下一条指令的地址，然后是函数的各个参数，在大多数编译器中，参数是由右往左入栈，然后是函数中的局部变量。注意，静态变量不入栈。出栈则刚好顺序相反。         堆：一般在堆的头部用一个字节存放堆的大小，具体内容由程序员安排。  <http://blog.chinaunix.net/attachment/201104/14/25119314_1302759660eYh5.png> | |