## 堆与栈的区别（内存和数据结构）

**数据结构中的堆与栈**

首先，堆与栈是两种数据结构：堆与栈。

堆与栈都是一种数据项按序排列的数据结构。

栈就像是装数据的桶子或箱子

它是一种具有后进先出性质的数据结构。

堆像一颗倒过来的树

堆是一种经过排序的树形数据结构，每个节点都有一个值。通常我们所说的堆的数据结构，是指二叉堆。堆常用来优先队列，对的存取是随意。

**内存分配中的堆与栈**

一般情况下程序放在Rom或Flash中，运行时需要拷贝到内存中执行。内存中的栈区处于相对较高的地址并以地址的增长方向为上的话，栈地址是向下增长的。栈中分配局部变量空间，堆区是向上增长的用于分配程序员申请的内存空间。另外还有静态区分配静态变量，全局变量空间的；只读区是分配常量和程序代码空间的；以及其他的一些分区。

来看一个网上很流行的经典例子

main.cpp

int a=0;//全局初始化区

char \*p1;//全局未初始化区

main()

{

int b;//栈

char s[]=”abc”;//栈

char\*p2;//栈

char \*p3=“1234567”；//1234567\0在常量区,p3在栈上

static int c=0;//全局（静态）初始化区

p1 = (char\*)malloc(10);//堆

p2 = (char\*)malloc(20);//堆

}

**堆与栈有以下几点不用**

1、申请方式与回收方式不同

栈区（Stack）--由编译器自动分配释放，存放函数的参数值，局部变量的值等。其操作方式方式类似于数据结构中的栈。

堆区（heap）--一般有程序员分配释放，若程序员不释放，程序结束时可能由OS回收。注意它与数据结构中的堆是两回事，分配方式类似于链表。

2、申请后系统的响应

栈：只要栈的剩余空间大于所申请空间，系统将为程序提供内存，否则将报异常提示栈溢出。

堆：首先应该知道操作系统有一个记录空闲内存地址的链表，当系统受到程序的申请时，会遍历该链表，寻找第一个空间大于所申请空间的堆。

3、申请大小的限制

在windows下，栈是向低地址扩展的数据结构，是一块连续的内存的区域。这句话的意思是栈顶地址和栈的最大容量是系统预先规定好的，在windows下，栈的大小是2M（也有人说是1M）。由于栈的大小有限，所以子函数还是有物理意义的，而不仅仅是逻辑意义。

**最后关于堆与栈的区别比喻**

引用一位前辈的看法：

使用栈就像我们去饭店吃饭，只管点菜（发出申请）、付钱和吃（使用），吃饱了就走，不必理会切菜、洗菜等准备工作和洗碗、刷锅等扫尾工作，它的好处就是快捷，但是自由度小。

使用堆就像是自己动手做喜欢吃的菜，比较麻烦，但是比较符合自己的口味，而且自由度大。

http://blog.csdn.net/wolenski/article/details/7951961