执行程序的内存分布总结

以下内容为各方资料汇总 所以逻辑顺序不大清晰

一般认为在c中分为这几个存储区：

1. 栈－－有编译器自动分配释放

2. 堆－－一般由程序员分配释放，若程序员不释放，程序结束时可能由OS回收

3. 全局区（静态区）－－ 全局变量和静态变量的存储是放在一块的，初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。程序结束释放。

4. 另外还有一个专门放常量的地方。程序结束释放。

在函数体中定义的变量通常是在栈上，用malloc, calloc, realloc等分配内存的函数分配得到的就是在堆上。在所有函数体外定义的是全局量，加了static修饰符后不管在哪里都存放在全局区（静态区）,在所有函数体外定义的static变量表示在该文件中有效，不能extern到别的文件用，在函数体内定义的static表示只在该函数体内有效。另外，函数中的"adgfdf"这样的字符串存放在常量区。比如：

int a = 0; //全局初始化区 数据段

char \*p1; //全局未初始化区 bss区

void main()

{

int b; //栈

char s[] = "abc"; //栈

char \*p2; //栈

char \*p3 = "123456"; //123456\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);

p2 = (char \*)malloc(20); //分配得来得10和20字节的区域就在堆区。

strcpy(p1, "123456"); //123456\0放在常量区，编译器可能会将它与p3所指向的"123456"优化成一块。

}

