<https://blog.csdn.net/xlhcgd/article/details/50814597>

c语言之 malloc函数详解

**一、原型：extern void \*malloc(unsigned int num\_bytes);**

头文件：#include <malloc.h> 或 #include <alloc.h> (注意：alloc.h 与 malloc.h 的内容是完全一致的。)

功能：分配长度为num\_bytes字节的内存块

说明：如果分配成功则返回指向被分配内存的指针，否则返回空指针NULL。

当内存不再使用时，应使用free()函数将内存块释放。

举例：

**[c-sharp]** [view plain](http://blog.csdn.net/xw13106209/article/details/4962479) [copy](http://blog.csdn.net/xw13106209/article/details/4962479)

1. #include<stdio.h>
2. #include<malloc.h>
3. **int** main()
4. {
5. **char** \*p;
7. p=(**char** \*)malloc(100);
8. **if**(p)
9. printf("Memory Allocated at: %x/n",p);
10. **else**
11. printf("Not Enough Memory!/n");
12. free(p);
13. **return** 0;
14. }

**二、函数声明(函数原型)：**

　　void \*malloc(int size);

　　说明：malloc 向系统申请分配指定size个字节的内存空间。返回类型是 void\* 类型。void\* 表示未确定类型的指针。C,C++规定，void\* 类型可以强制转换为任何其它类型的指针。这个在MSDN上可以找到相关的解释，具体内容如下：

**malloc** returns a void pointer to the allocated space, or **NULL** if there is insufficient memory available. To return a pointer to a type other than **void**, use a type cast on the return value. The storage space pointed to by the return value is guaranteed to be suitably aligned for storage of any type of object. If size is 0, **malloc** allocates a zero-length item in the heap and returns a valid pointer to that item. Always check the return from **malloc**, even if the amount of memory requested is small.

**三、malloc与new的不同点**

　　从函数声明上可以看出。malloc 和 new 至少有两个不同: ***new 返回指定类型的指针，并且可以自动计算所需要大小。***比如：

      int \*p;

　　p = new int; //返回类型为int\* 类型(整数型指针)，分配大小为 sizeof(int);

　　或：

　　int\* parr;

　　parr = new int [100]; //返回类型为 int\* 类型(整数型指针)，分配大小为 sizeof(int) \* 100;

***而 malloc 则必须由我们计算要字节数，并且在返回后强行转换为实际类型的指针。***

   　int\* p;

　　p = (int \*) malloc (sizeof(int));

　　第一、malloc 函数返回的是 void \* 类型，如果你写成：p = malloc (sizeof(int)); 则程序无法通过编译，报错：“不能将 void\* 赋值给 int \* 类型变量”。所以必须通过 (int \*) 来将强制转换。

　　第二、函数的实参为 sizeof(int) ，用于指明一个整型数据需要的大小。如果你写成：

　　int\* p = (int \*) malloc (1);

　　代码也能通过编译，但事实上只分配了1个字节大小的内存空间，当你往里头存入一个整数，就会有3个字节无家可归，而直接“住进邻居家”！造成的结果是后面的内存中原有数据内容全部被清空。

　　malloc 也可以达到 new [] 的效果，申请出一段连续的内存，方法无非是指定你所需要内存大小。

　　比如想分配100个int类型的空间：

　　int\* p = (int \*) malloc ( sizeof(int) \* 100 ); //分配可以放得下100个整数的内存空间。

　　另外有一点不能直接看出的区别是，malloc 只管分配内存，并不能对所得的内存进行初始化，所以得到的一片新内存中，其值将是随机的。

　　除了分配及最后释放的方法不一样以外，通过malloc或new得到指针，在其它操作上保持一致。

**总结：**

malloc()函数其实就在内存中找一片指定大小的空间，然后将这个空间的首地址范围给一个指针变量，这里的指针变量可以是一个单独的指针，也可以是一个数组的首地址，这要看malloc()函数中参数size的具体内容。我们这里malloc分配的内存空间在逻辑上连续的，而在物理上可以连续也可以不连续。对于我们程序员来说，我们关注的是逻辑上的连续，因为操作系统会帮我们安排内存分配，所以我们使用起来就可以当做是连续的。