**VC调试中出现的一些内存值**

调试过debug版本的vc程序的人一定对0xCCCCCCCC和0xCDCDCDCD这样的内存很有印象。这是debug版本的CRT为了方便调试程序，在分配出来还没有初始化的时候提供的初始值。

实际上，Windows上面还有更多这样的初始值，见下表：

|  |  |
| --- | --- |
| Uninitialized |  |
| BAADF00D | Used by Microsoft's LocalAlloc/GlobalAlloc/HeapAlloc(LMEM\_FIXED) to mark uninitialised allocated heap memory |
| CCCCCCCC | Used by Microsoft's C++ debugging runtime library to mark uninitialised stack memory |
| CDCDCDCD | Used by Microsoft's C++ debugging runtime library to mark uninitialised heap memory |
|  |  |
| Freed |  |
| FEEEFEEE | Used by Microsoft's LocalFree/GlobalFree/HeapFree() to mark freed heap memory |
| DDDDDDDD | Used by MicroQuill's SmartHeap and Microsoft's C++ debugging heap to mark freed heap memory |
|  |  |
| No man's land |  |
| ABABABAB | Used by Microsoft's LocalAlloc/GlobalAlloc/HeapAlloc() to mark "no man's land" guard bytes after allocated heap memory |
| FDFDFDFD | Used by Microsoft's C++ debugging heap to mark "no man's land" guard bytes before and after allocated heap memory |

表格来源：<http://en.wikipedia.org/wiki/Magic_number_(programming)>

除了debug CRT会帮你添加这些初始化值之外，微软的堆管理函数也会在分配和释放的时候添加一些初始化值。

LocalAlloc/GlobalAlloc，如果指定的是LMEM\_FIXED（默认就指定了这个），并且没有指定LMEM\_ZEROINIT，则分配的内存中初始化值为BAADF00D（可以理解成badfood，也就是不能直接吃的意思，呵呵）。调用LocalFree/GlobalFree则其值会变为FEEEFEEE）可以理解成Free）。

HeapAlloc只要没有指定HEAP\_ZERO\_MEMORY，也是一样初始化值是BAADF00D，HeapFree之后则是FEEEFEEE。

另外，windows的三个heap分配函数（LocalAlloc/GlobalAlloc/HeapAlloc）分配内存的时候，会在分配的内存后面另外添加8个byte的Guard数据，也就是上表中的No man's land。其数值为ABABABAB

关于windows的heap管理函数，还有以下两点需要注意：

1. 如果分配的内存过大（比如几十M，HeapAlloc会转换成用VirtualAlloc来分配，而不是从堆上进行分配，所以分配和释放之后的内存值不是上表所示。
2. 开启pageheap之后，调用HeapAlloc分配的初始化值为0xc0c0c0c0。HeapFree之后的内存则无法访问

如果你的内存管理用的是new（malloc）和delete（free）这样的CRT函数，那么情况会更复杂一些。这些CRT内存管理函数是建立在上面的windows heap管理函数之上的。

尤其是debug版本的CRT，会做更多的事情，详细见<http://www.nobugs.org/developer/win32/debug_crt_heap.html>

这里简要说明一下，new（malloc）分配的未初始化内存的值为CDCDCDCD，delete（free）之后的未初始化值为DDDDDDDD。

另外 ，debug CRT也有跟windows 的 heap管理函数一样的No man's land数据，他们是FDFDFDFD（可以理解成fence），总共8个bytes，4个byte在payload前面 ，4个byte在payload后面。  
需要注意的是CRT的管理数据实际上也是windows的heap管理函数的payload，所以当我们用CRT的函数来分配内存时，比如说10bytes，CRT向windows的heap管理函数会需要申请额外的内存（大概是40byte，也就是说总共50byte）。这额外的内存是CRT用来管理CRT内存的，其中就包括No man's land数据，另外还有分配了多少内存，内存类型，调用分配函数时的文件名和行数等等。详细这40bytes作和作用请参见上面的URL

这里还有一些关于Debug CRT的更详细的介绍：<http://www.codeguru.com/cpp/w-p/win32/tutorials/article.php/c9535/Inside-CRT-Debug-Heap-Management.htm>

值得注意的是，以上是Debug CRT才会有的操作，release版本的CRT是直接调用windows的heap manager函数，所以其表现跟用户直接调用heap manager函数是一样的