# 学习大纲

|  |
| --- |
|  |

# 演练

## 1.新建一共cpp常规空项目,取名Lesson86-mem-data-op,然后新建一共cpp源文件,取名:mem-data-op.cpp,添加一下检查代码,然后不要运行,看看能否正常运行

|  |
| --- |
|  |
|  |

### 正常

## 2.下面是练习代码

|  |
| --- |
| #include<Windows.h>  #include<stdio.h>  #include<stdlib.h>  #define MEMBLOCKSIZE 32  BOOL ShowMemContent(LPVOID lpMem,SIZE\_T size)  {  BYTE lpShow[MEMBLOCKSIZE];  INT i=0;  CopyMemory(lpShow,lpMem,size);  if(size > MEMBLOCKSIZE)  {  printf("Over Flow!!!\n");  return FALSE;  }  for(;i<size;i++)  {  printf("%0.2X ",lpShow[i]);  if(!(i+1)%16)  {  printf("\n");  }  }  printf("\n");  return TRUE;  }  int main()  {  HANDLE hHeap;  LPVOID lpMem1,lpMem2;  hHeap = GetProcessHeap();//在进程的默认堆上面操作  lpMem1 = HeapAlloc(hHeap,HEAP\_NO\_SERIALIZE,MEMBLOCKSIZE);  ShowMemContent(lpMem1,MEMBLOCKSIZE);//没有初始化,显示的是垃圾数据  lpMem2 = HeapAlloc(hHeap,HEAP\_ZERO\_MEMORY,MEMBLOCKSIZE);//这个标志表示创建后马上用0初始化  ShowMemContent(lpMem2,MEMBLOCKSIZE);//初始化为0,显示的都是0  //初始化第一块内存  ZeroMemory(lpMem1,MEMBLOCKSIZE);  ShowMemContent(lpMem1,MEMBLOCKSIZE);//初始化为0,显示的都是0  //也可以用指定的数据填充内存  FillMemory(lpMem1,MEMBLOCKSIZE,0xAA);  ShowMemContent(lpMem1,MEMBLOCKSIZE);//填充为0xAA,显示的都是AA  FillMemory(lpMem2,MEMBLOCKSIZE/2,0xBB);  ShowMemContent(lpMem2,MEMBLOCKSIZE);//填充一半为0xBB,显示的一半是BB  //移动内存的内容  MoveMemory(lpMem2,lpMem1,MEMBLOCKSIZE);//其实就拷贝和CopyMemory函数是一样的.  ShowMemContent(lpMem2,MEMBLOCKSIZE);  ShowMemContent(lpMem1,MEMBLOCKSIZE);  //释放内存  HeapFree(hHeap,0,lpMem1);  HeapFree(hHeap,0,lpMem2);  system("pause");  return 0;  } |

# 这一节学习到此为止,是比较简单的API实例