What is difference between BaseAddress and AllocationBase in MEMORY_BASIC_INFORMATION struct?

Asked 11 years, 2 months ago Modified 11 years, 2 months ago Viewed 4k times

9

In MSDN i find following`

BaseAddress - A pointer to the base address of the region of pages.

AllocationBase - A pointer to the base address of a range of pages allocated by the VirtualAlloc function. The page pointed to by the BaseAddress member is contained within this allocation range.

But i don't understand what is difference really. Can anyone tell me difference? (not like in MSDN :))

windows windows-7 process operating-system msdn

Share Improve this question Follow

asked Oct 19, 2013 at 10:49
know_everything
123 ● 2 ● 7

1 Answer

Sorted by: Highest score (default)

23

Virtual memory allocations on Windows are made with a granularity of 64 kilobytes, the value of SYSTEM_INFO.dwAllocationGranularity. But virtual memory pages are 4096 bytes, the value of SYSTEM_INFO.dwPageSize.

When you allocate virtual memory with VirtualAlloc, you'll always get a chunk back whose BaseAddress equals AllocationBase. But if you then alter the page protection of one or more of the pages within this chunk then you can observe this chunk being subdivided with a different BaseAddress. Best shown with a sample program, run this on MSVC++:

```
#include "stdafx.h"
#include <Windows.h>
#include <stdio.h>
#include <conio.h>

void showmem(void* mem) {
    MEMORY_BASIC_INFORMATION info = {};
    VirtualQuery(mem, &info, sizeof info);
    printf("Alloc = %p, base = %p, size = %d, protect = %d\n",
           info.AllocationBase, info.BaseAddress, info.RegionSize, info.Protect);
}


int main() {
    BYTE* mem = (BYTE*)VirtualAlloc(0, 65536, MEM_COMMIT, PAGE_READWRITE);
    printf("%s", "Initial allocation:\n");
    showmem(mem);

    DWORD oldprotect;
    BOOL ok = VirtualProtect(mem + 4096, 4096, PAGE_NOACCESS, &oldprotect);
    printf("%s", "\nAfter protection changes:\n");
    showmem(mem);
    showmem(mem + 4096);
    showmem(mem + 4096 + 4096);

    _getch();
    return 0;
}
```

Sample output of this program:

```
Initial allocation:
Alloc = 00ED0000, base = 00ED0000, size = 65536, protect = 4

After protection changes:
Alloc = 00ED0000, base = 00ED0000, size = 4096, protect = 4
Alloc = 00ED0000, base = 00ED1000, size = 4096, protect = 1
Alloc = 00ED0000, base = 00ED2000, size = 57344, protect = 4
```

And note how the VirtualProtect() call required the original chunk to be split in 3 regions with different BaseAddress but the same AllocationBase.