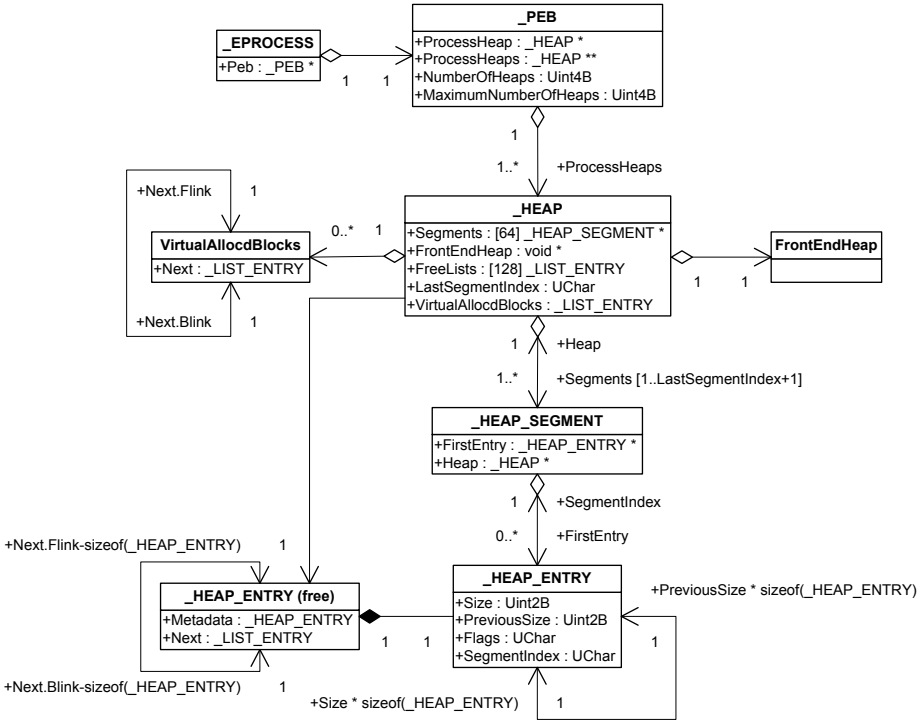
05 (0X05): HEAP / POOL / PAGED / NONPAGED

Heap and pool are kind of dynamic memory where the former term is usually used in user space and the latter in kernel mode and space parlance.

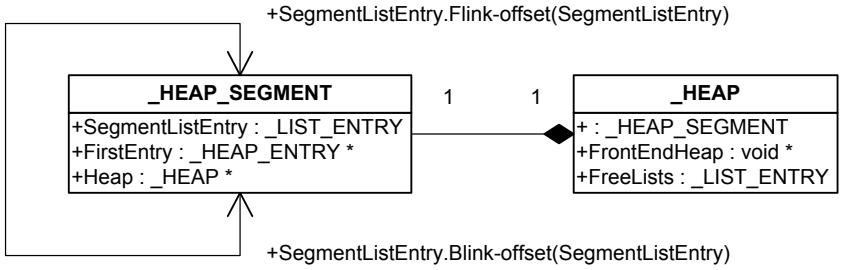
Paged pool pages can be moved or being in transition to page file(s) and therefore reading or writing them at IRQL 2 or above may result in bugcheck 0xA (Vostokov, Bugchecks depicted: IRQL_NOT_LESS_OR_EQUAL). Non-paged pool pages are always present in physical memory. Heap can be non-paged too if locked in memory.

Excellent description of process heap and associated issues like heap corruption can be found in Advanced Windows Debugging book (Hewardt & Pravat, 2008). Here is the corresponding UML static structure diagram for Windows XP/2003 heap implementation (reengineered using **dt** WinDbg command) where only details useful to understand heap structure and its navigation are included:



Large heap allocations beyond certain threshold go directly to virtual memory manager (Vostokov, Large Heap Allocations).

There are minor changes in Vista and Windows Server 2008. The most noticeable of them is the fact that heap segments and free lists are now doubly-linked lists instead of fixed arrays. The first heap segment is at the beginning of the _HEAP structure and additional heap segments are linked through _HEAP_SEGMENT.SegmentListEntry:



For heap and pool corruption crash dump analysis patterns please refer to (Vostokov, Dynamic Memory Corruption (process heap)) and (Vostokov, Dynamic Memory Corruption (kernel pool)). For double free analysis please refer to (Vostokov, Double Free (process heap)) and (Vostokov, Double Free (kernel pool)).

Uncontrolled growth of heap or pool size is called heap or pool memory leak. For related patterns and debugging techniques please refer to (Vostokov, Insufficient Memory (committed memory)), (Vostokov, Insufficient Memory (handle leak)), (Vostokov, Insufficient Memory (kernel pool)), (Vostokov, Memory Leak (process heap)) and (Vostokov, Memory Leak (.NET heap)).

By default, kernel pool allocations since Windows Server 2003 are marked with 4-byte tags to help in driver identification. On Windows XP you need to enable this feature using gflags.exe. Anonymous pool allocations are marked with 'None' tag. For further information please refer to the case study about 'Ddk' tag. (Vostokov, The search for 'Ddk' tag).