内核模式堆(系统内存池)

系统初始化时，内存管理器会创建两个动态大小的内存池(也叫作堆)，大部分内核模式组件借此来分配系统内存。

1. **非换页池**。由一系列可保证在任何时间总是驻留在物理内存中的系统虚拟地址组成。因此任何时间都可在不产生页面错误的情况下访问，这也意味着可以从任何IRQL访问。需要非换页池的原因之一在于，在 DPC/dispatch 级别或更高级别上，页面错误无法被满足。因此任何代码和数据如果需要在DPC/dispatch 级别或更高级别上执行或访问都必须位于非换页池内存中。

当系统试图访问一个尚未加载到物理内存中的内存页面时，会发生页面错误，之后会发生换页操作，把磁盘中的内存页换到物理内存中，但是换页操作的IRQL级别很低（PASSIVE\_LEVEL）

1. **换页池**。一种位于系统空间中的虚拟内存区域，可以换入或换出系统。如果设备驱动程序无须从 DPC/dispatch 级别或更高级别访问内存，即可使用换页池。换页池可从任何进程上下文中访问。

这两个内存池都位于系统地址空间中，可映射到每个进程的虚拟地址空间。执行体提供了从这些池中分配和撤销分配所需的例程。用名称以 ExAllocatePool、ExAllocatePoolWithTag 以及 ExFreePool开头的函数

系统启动时会包含4个换页池(组合在一起形成完整的系统换页池)和两个非换页池。根据系统中 NUMA 节点的数量，还可以创建更多换页池，最多可达 64 个。使用超过一个换页池可以降低系统代码在并发调用池例程时被阻塞的频率。此外所创建的不同池会映射至不同的虚拟地址范围，并对应系统中不同的NUMA节点。用于描述池分配情况的不同数据结构(如大页面旁视表)也会映射给不同的NUMA节点。

除了换页和非换页池，还有一些包含特殊属性或提供特殊用途的池。例如，在会话空间中有一个池区域，可用于存储该会话中所有进程通用的数据。来自其他池的分配也叫作特殊池 (special pool)，这种池的两边会围绕着标记为“不可访问”的页面，借此可对该内存池区域分配前后访问内存的代码中出现的问题进行隔离

特殊池（Special Pool）是 Windows 操作系统内核中一种用于调试目的的内存池。特殊池的主要用途是检测内存泄漏、越界访问和其他内存相关的问题。当开发人员怀疑某个驱动程序或内核组件中存在这类问题时，特殊池可以帮助定位和诊断这些问题。

特殊池的主要特点是在每个内存分配的前后添加“不可访问”的内存页面，这样如果代码试图访问这些页面，操作系统就会立即捕获到访问违规并生成错误报告。这有助于发现内存越界访问、双重释放和其他内存相关的错误。

要启用特殊池，可以在注册表中进行配置，或者使用 Global Flags 工具（gflags.exe）。以下是在注册表中启用特殊池的路径：

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management

在右侧窗格中，创建或修改名为 SpecialPool 的 DWORD 值。将其值设置为 1 以启用特殊池。重启计算机使更改生效。

在启用特殊池后，操作系统会在内核池分配时自动应用特殊池。请注意，特殊池会增加内存使用和性能开销，因此在正常运行时应该禁用特殊池，仅在调试时启用。

池的大小

非换页池的初始大小取决于系统中的物理内存数量，并会按需增长。对于非换页池其初始大小为系统物理内存容量的 3%。但如果这个值小于 40MB，只要物理内存数量的10%超过40MB，系统将一直使用40MB的大小。否则将使用物理内存数量的10%作为最小值。Windows 会动态地选择内存池的最大大小，并允许特定内存池从初始大小增大至下表所列出的最大值。

|  |  |  |  |
| --- | --- | --- | --- |
| **内存池类型** | **32位系统最大值** | **64位系统最大值(Windows8、Windows Server 2012 )** | **64位系统最大值(Windows8.1、Windows 10、Windows Server 2012R2、Windows Server 2016 )** |
| 非换页池 | 物理内存的75%或2GB，两者间取较小值 | 物理内存的75%或128GB，两者间取较小值 | 16TB |
| 换页池 | 2GB | 384GB | 15.5TB |

上述计算而来的值中，有4个会存储在 Windows 8.x和Windows Server 2012/R2的内核变量中，其中3个可通过性能计数器的方式暴露，另一个仅作为性能计数器的值进行计算。Windows 10和Windows Server 2016将这些全局变量转移到了全局内存管理结构(MI\_SYSTEM\_INFORMATION)中名为 MiState 的字段中。在该字段中，这些信息存储在一个名为Vs(类型为\_MIVISIBLE STATE)的变量内部。全局变量 MiVisibleState 也可指向该Vs成员。这些变量和计数器见下表

|  |  |  |
| --- | --- | --- |
| **内核变量** | **性能计数器** | **描述** |
| MmSizeOfNonPagedPoolInBytes | Memory: Pool nonpaged bytes | 非换页池的初始大小。根据内存需求变化，系统可自动减小或增大该值。内核变量不会显示这些变化，但性能计数器可以显示 |
| MmMaximumNonPagedPoolInBytes( Windows 8.x 和Windows Server 2012/R2) | 不适用 | 非换页池的最大大小 |
| MiVisibleState->MaximumNonPagePool InBytes(Windows 10和 Windows Server 2016) | 不适用 | 非换页池的最大大小 |
| 不适用 | Memory: Pool paged bytes | 换页池虚拟大小的当前总和 |
| MmPagedPoolWs结构(类型为MMSUPPORT)中的 WorkingSetSize(页面数量)(Windows 8.x和Windows Server 2012/R2) | Memory: Pool paged resident bytes | 换页池的当前物理（驻留） |
| MmSizeOfPagedPoolInBytes ( Windows 8.x 和Windows Server 2012/R2) | 不适用 | 换页池的最大（虚拟）大小 |
| MiState.VsSizeOfPagedPoolln Bytes  ( Windows 10和 Windows Server 2016) | 不适用 | 换页池的最大（虚拟）大小 |

旁视列表

Windows 提供了一种名为旁视列表 (look-aside list)的快速内存分配机制。内存池和旁视列表的差异在于：一般来说内存池分配的大小可能各异，但旁视列表只包含固定大小的块。从所提供的功能来看，一般的内存池更为灵活，但旁视列表由于不使用任何自旋锁速度更快一些。

当需要频繁分配和释放大小固定的内存块时，使用旁视列表可以提高性能。旁视列表将已分配但未使用的内存块保存在一个链表中。当需要分配内存时，系统首先检查旁视列表中是否有可用的内存块。如果有，则直接从旁视列表中获取，而不需要重新分配内存。当释放内存时，系统将内存块放回旁视列表中，以便后续重新使用。

在Windows内核编程中，可以使用以下函数来操作旁视列表：

ExInitializeNPagedLookasideList：用于初始化一个非分页旁视列表。

ExInitializePagedLookasideList：用于初始化一个分页旁视列表。

ExAllocateFromNPagedLookasideList：从非分页旁视列表中分配一个内存块。

ExAllocateFromPagedLookasideList：从分页旁视列表中分配一个内存块。

ExFreeToNPagedLookasideList：将一个内存块归还到非分页旁视列表。

ExFreeToPagedLookasideList：将一个内存块归还到分页旁视列表。

ExDeleteNPagedLookasideList：删除一个非分页旁视列表，释放其所有资源。

ExDeletePagedLookasideList：删除一个分页旁视列表，释放其所有资源。

如果某个旁视列表是空的(例如在首次创建时)，系统必须从换页或非换页池中分配但如果其中包含已释放的块，内存分配请求将以非常快的速度满足。(随着块被逐渐归还该列表也会增长。)内存池分配例程会根据设备驱动程序或执行体子系统从该列表中分配的频率，自动优化旁视列表所存储的已释放块的数量。分配频率越高，列表中存储的块数量也就越多。如果不再据此进行分配，旁视列表的大小还会自动减小。(这种检查每秒进行一次，此时将唤醒平衡集管理器系统线程并调用ExAdjustLookasideDepth函数。