内存管理

支持模块化程序设计的计算环境和数据的灵活使用，可以很好地满足用户的需求。系统管理员需要有效且有条理控制存储器分配。为满足这些要求，操作系统担负着5项存储器管理职责:

* **进程隔离:** 操作系统必须保护独立的进程，防止相互干扰各自的存储空间，包括数据和指令。
* **自动分配和管理:** 程序应该根据需要在存储层次间动态地分配，分配对程序员是透明的。因此，程序员无须关心与存储限制有关的问题，操作系统会有效地实现分配问题，可仅在需要时才给程序额外分配存储空间。
* **支持模块化的程序设计:** 程序员应该能够定义程序模块，并动态地创建、销毁模块，动态地改变模块的大小。
* **保护和访问控制:** 不论在存储层次中的哪一级，存储器的共享都会产生一个程序访问另一个程序存储空间的潜在可能性。当某个特定的应用程序需要共享时，这是可取的。但在其他时候，它可能会威胁到程序的完整性，甚至威胁到操作系统自身。操作系统必须允许一部分内存可以由各种用户以各种方式进行访问。
* **长期存储:** 许多应用程序需要在计算机关机后长时间地保存信息。

通常，操作系统使用虚拟内存和文件系统机制来满足这些要求。

虚拟内存允许程序以逻辑方式访问存储器，而不考虑物理内存上可用的空间数量。虚拟内存的构想是为了满足有多个用户作业同时驻留在内存中的要求，因此在一个进程被写出到辅助存储器中且后续进程被读入时，连续的进程执行之间将不会脱节。进程大小不同时，若处理器在很多进程间切换，则很难把它们紧密地压入内存，因此人们引入了分页系统(分页内存)。程序通过**虚地址**(virtual address)访问内存，虚地址由页号和页中的偏移量组成。进程的每页都可置于内存中的任何地方，分页系统提供了程序中使用的虚地址和内存中的**实地址**(real address)或物理地址之间的动态映射。

有了动态映射硬件后，还需要消除一个进程的所有页同时驻留在内存中的要求。一个进程的所有页都保留在磁盘中，进程执行时，一部分也会调入内存中。若需要访问的某页不在内存中，会触发缺页(missing page)操作。这一配置称为**虚拟内存**(virtual memory)。