

虚拟内存

一个系统中的进程是与其他进程共享 CPU 和主存资源的。然而，共享主存会形成一些特殊的挑战。随着对 CPU 需求的增长，进程以某种合理的平滑方式慢了下来。但是如果太多的进程需要太多的内存，那么它们中的一些就根本无法运行。当一个程序没有空间可用时，那就是它运气不好了。内存还很容易被破坏。如果某个进程不小心写了另一个进程使用的内存，它就可能以某种完全和程序逻辑无关的令人迷惑的方式失败。

为了更加有效地管理内存并且少出错，现代系统提供了一种对主存的抽象概念，叫做虚拟内存 (VM)。虚拟内存是硬件异常、硬件地址翻译、主存、磁盘文件和内核软件的完美交互，它为每个进程提供了一个大的、一致的和私有的地址空间。通过一个很清晰的机制，虚拟内存提供了三个重要的能力：1) 它将主存看成是一个存储在磁盘上的地址空间的高速缓存，在主存中只保存活动区域，并根据需要在磁盘和主存之间来回传送数据，通过这种方式，它高效地使用了主存。2) 它为每个进程提供了一致的地址空间，从而简化了内存管理。3) 它保护了每个进程的地址空间不被其他进程破坏。

虚拟内存是计算机系统最重要的概念之一。它成功的一个主要原因就是因为它是沉默地、自动地工作的，不需要应用程序员的任何干涉。既然虚拟内存幕后工作得如此之好，为什么程序员还需要理解它呢？有以下几个原因：

- 虚拟内存是核心的。虚拟内存遍及计算机系统的所有层面，在硬件异常、汇编器、链接器、加载器、共享对象、文件和进程的设计中扮演着重要角色。理解虚拟内存将帮助你更好地理解系统通常是如何工作的。
- 虚拟内存是强大的。虚拟内存给予应用程序强大的能力，可以创建和销毁内存片 (chunk)、将内存片映射到磁盘文件的某个部分，以及与其他进程共享内存。比如，你知道可以通过读写内存位置读或者修改一个磁盘文件的内容吗？或者可以加载一个文件的内容到内存中，而不需要进行任何显式地复制吗？理解虚拟内存将帮助你利用它的强大功能在应用程序中添加动力。
- 虚拟内存是危险的。每次应用程序引用一个变量、间接引用一个指针，或者调用一个诸如 malloc 这样的动态分配程序时，它就会和虚拟内存发生交互。如果虚拟内存使用不当，应用将遇到复杂危险的与内存有关的错误。例如，一个带有错误指针的程序可以立即崩溃于“段错误”或者“保护错误”，它可能在崩溃之前还默默地运行了几个小时，或者是最令人惊慌地，运行完成却产生不正确的结果。理解虚拟内存以及诸如 malloc 之类的管理虚拟内存的分配程序，可以帮助你避免这些错误。

这一章从两个角度来看虚拟内存。本章的前一部分描述虚拟内存是如何工作的。后一部分描述的是应用程序如何使用和管理虚拟内存。无可避免的事实是虚拟内存很复杂，本章很多地方都反映了这一点。好消息就是如果你掌握这些细节，你就能够手工模拟一个小系统的虚拟内存机制，而且虚拟内存的概念将永远不再神秘。

第二部分是建立在这种理解之上的，向你展示了如何在程序中使用和管理虚拟内存。你将学会如何通过显式的内存映射和对像 malloc 程序这样的动态内存分配器的调用来管