

Linux内核按需动态装载和卸掉模块 (1) PDF转换可能丢失
图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E5_86_85_E6_A0_c103_144633.htm Linux是单内核结构，也就是说，它是一个大程序，其中任一函数都可以访问公共数据结构和其它函数调用。(作为操作系统)另外一种可能的结构是多核式的，各功能块自成一体，相互之间由严格的通信机制相连。单核结构在添加新模块时，一种方法是重新调整设置，所以非常费时。比如，你想在内核中加一个NCR 810 SCSI的驱动程序，你必须重新设置，重建内核。这也有另外一个办法，Linux 允许动态装载和卸掉模块。Linux 模块是一段可以在机器起动后任意时间被动态连接的代码。在不需要时，它们可以被从内核中卸掉。大多数Linux 模块是设备驱动程序或伪设备驱动程序，如网络驱动程序，文件系统等。你可以使用 `insmod` 和 `rmmod` 命令来装载和卸掉 Linux 模块，内核自己也可以调用内核驻留程序(Kernel) 来按需要装载和卸掉模块。按需动态装载模块可以使内核保持最小，并更具灵活性。我现在的 Intel 内核由于大量使用动态装载模块，只有 406 K 字节。例如，我很少用到 VFAT 文件系统，所以我让 Linux 内核只在我装载 VFAT 分区时，才自动上载 VFAT 文件系统。当我卸掉 VFAT 分区时，内核会检测到，并自动卸掉 VFAT 文件系统。当测试新程序时，你如果不想每次都重建内核，动态装载模块是非常有用的。但是，运用模块会多消耗一些内存，并对速度有一定影响。并且模块装载程序是一段代码，它的数据将占用一部份内存。这样还会造成不能直接访问内核资源，效率不高的问题。一旦 Linux 模块被装载后，它就和一般内核代码一样，对其

它内核代码,享受同样的访问权限。换句话说, Linux 内核模块可以像其它内核代码,或驱动程序一样使系统崩溃。模块可以使用内核资源,但首先它需知道怎样调用。例如,一个模块要调用 `Kmalloc()` (内核内存分配程序)。但在模块建立时,它并不知道到哪儿去找 `Kmalloc()`,所以在它被装载时,内核必须先设定模块中所有 `Kmalloc()` 调用的函数指针。内核有一张所有资源调用的列表,在模块被装载时,内核重设所有资源调用的函数指针。Linux 允许栈式模块,即一个模块调用另一个模块的函数。例如,由于 VFAT 文件系统可以看成是 FAT 文件系统的超集,所以 VFAT 文件系统模块需要调用 FAT 文件系统提供的服务。一个模块调用另一模块的资源与调用内核资源很相似。唯一的不同是被调用的模块需被先载入。一个模块被载入后,内核将修改它的内核符号表(KERNEL SYMBOL TABLE),加入新载入模块提供的所有资源和符号。所以另一个模块被载入时,它就可以调用所有已载入模块提供的服务。当卸掉一模块时,内核先确定该模块不会再被调用,然后通过某种方式通知它。在该模块被内核卸掉以前,该模块须释放所有占用的系统资源。例如,内存或中断,当模块被卸掉后,内核从内核符号表中删除所有该模块提供的资源。

100Test 下载频道开通,各类考试题目直接下载。详细请访问
www.100test.com