

将OS_2应用程序移植到Linux操作系统（2）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__E5_B0_86OS_2_E5_BA_94_c103_144625.htm 队列 OS/2 队列是一个允许进程把内存指针传给另一个进程的机制。传送进程必须通过 DosGiveSharedMemory 给接受进程提供到内存段的访问，然后把内存在段中的位置传送给接收进程。像机制名暗示的一样，队列可以在等待另一个进程读取它们的同时保存大量的这些内存位置。Linux 没有这个概念。在可能的解决方案中，其中之一就是使用 System V 共享内存段。System V 共享内存段的问题是限制了被允许的段的数量。另一个方法是使用文件映射并使用控制结构，该结构设置元素（内存位置）的顺序和对文件的访问权限。信号量（Semaphore）OS/2 提供三种主要类型的信号量：事件、互斥和多等待。Linux 提供一个基于 System V IPC 的信号量机制并且支持信号量作为 pthread 库的一部分。OS/2 信号量是一个单独的实体，给您提供一个定义良好的行为。然而，可以将 System V 信号量定义成几组，这是可配置的。在不重新编译内核的情况下，每个组最多可以有 250 个信号量（2.4 内核）。这种系统在系统内只能有一定数量的信号量组（可以通过重编译内核来更改这个数量）。System V 信号量基本上是计数变量，您可以增加或减少这些变量，并且这为您模仿事件信号量和互斥信号量的行为提供了足够的功能。唯一的问题是执行定时等待。在 Linux 上，您只能尝试 / 等待或等待。不允许超时。为了模仿超时，您需要构建某种定时机制，该机制能发送一个信号来中断正在等待一个事件的线程。多等待信号量是一个已定义

信号量的集合，这些信号量可以作为一组等待。System V 机制利用组，并且能等待许多与同一个组相关的操作。然而，只有应用程序仔细地规划了它对信号量（多等待中的所有信号量都来自于同一个信号量组）的使用，您才能得到期望的结果。我们实现了一个机制，它使线程能够等候这组内的每个信号量来模拟这一行为。pthread 库提供了能合理模仿 OS/2 行为的信号量和互斥（以及条件变量）。然而，还存在缺陷。请注意，pthread 信号量和互斥是不可共享的。换言之，仅在那个进程中保证它们。如果您的应用程序仅使用私有信号量，或者如果没有其它进程需要访问它的信号量，那么使用 pthread 互斥和信号量可能是最好的计划。但是，如果您的应用程序共享这些 IPC 机制，那么您需要用 System V 机制、共享内存和线程来实现这些机制。信号在 OS/2 上，为了指明某些错误情况，要抛出异常。在 Linux 上，这些被称为信号。与异常一样，除了 SIGKILL 和 SIGSTOP 之外，信号都可以被捕获。在 OS/2 上，您可以定义一系列在接收到异常时需要执行的函数。应用程序可以在 OS/2 上定义不止一个异常处理程序。在 Linux 上，因为函数被覆盖了，所以您只能定义一个异常处理程序。管道在 OS/2 上，管道是双向的，而 Linux 上的管道是单向的。为了模仿 OS/2 管道的行为，您需要创建两个管道，通过一个文件句柄来索引。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