

Linux下关于互斥锁及同步的移植Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_Linux\\_E4\\_B8\\_8B\\_E5\\_85\\_c103\\_645031.htm](https://www.100test.com/kao_ti2020/645/2021_2022_Linux_E4_B8_8B_E5_85_c103_645031.htm) 将win32程序关于多线程的内容移植到Linux下面，不能简单的按照函数对应来移植。不过通过下面的对应关系，再加上你对这些模式的深入了解，相信会移植的很成功。信号量 Windows 信号量是一些计数器变量，允许有限个线程/进程访问共享资源。Linux POSIX 信号量也是一些计数器变量，可以用来在Linux上实现Windows上的信号量功能。信号量的类型：Windows提供了有名(named)信号量和无名(unnamed)信号量。有名信号量可以在进程之间进行同步。在Linux上，在相同进程的不同线程之间，则只使用POSIX信号量。在进程之间，可以使用System V信号量。等待函数中的超时：当在一个等待函数中使用时，可以为Windows信号量对象指定超时值。在Linux中，并没有提供这种功能，只能通过应用程序逻辑处理超时的的问题。创建信号量在Windows中，可以使用CreateSemaphore()创建或打开一个有名或无名的信号量。HANDLE CreateSemaphore(LPSECURITY\_ATTRIBUTES lpSemaphoreAttributes, LONG lInitialCount, LONG lMaximumCount, LPCTSTR lpName)。在这段代码中：lpSemaphoreAttributes是一个指向安全性属性的指针。如果这个指针为空，那么这个信号量就不能被继承。lInitialCount是该信号量的初始值。lMaximumCount是该信号量的最大值，该值必须大于0。lpName是信号量的名称。如果该值为NULL，那么这个信号量就只能在相同进程的不同线程之间共享。否则，就可以在不同的进程之间进行共享。

这个函数创建信号量，并返回这个信号量的句柄。它还将初始值设置为调用中指定的值。这样就可以允许有限个线程来访问某个共享资源。在 Linux 中，可以使用 `sem_init()` 来创建一个无名的 POSIX 信号量，这个调用可以在相同进程的线程之间使用。它还会对信号量计数器进行初始化：`int sem_init(sem_t *sem, int pshared, unsigned int value)`。在这段代码中：`value`(信号量计数器)是这个信号量的初始值。`pshared`可以忽略，因为在目前的实现中，POSIX 信号量还不能在进程之间进行共享。这里要注意的是，最大值基于 `semaphore.h` 头文件中定义的 `SEM_VALUE_MAX`。在 Linux 中，`semget()` 用于创建 System V 信号量，它可以在不同集成的线程之间使用。可以用它来实现与 Windows 中有名信号量相同的功能。这个函数返回一个信号量集标识符，它与一个参数的键值关联在一起。当创建一个新信号量集时，对于与 `semid_ds` 数据结构关联在一起的信号量，`semget()` 要负责将它们进行初始化，方法如下：`sem_perm.cuid` 和 `sem_perm.uid` 被设置为调用进程的有效用户 ID。`sem_perm.cgid` 和 `sem_perm.gid` 被设置为调用进程的有效组 ID。`sem_perm.mode` 的低 9 位被设置为 `semflg` 的低 9 位。`sem_nsems` 被设置为 `nsems` 的值。`sem_otime` 被设置为 0。`sem_ctime` 被设置为当前时间。

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](http://www.100test.com)