

# 授客的博客

http://blog.sina.com.cn/ishouke  [订阅]  [手机订阅]

首页    博文目录    图片    关于我

个人资料



授客

微博

加好友

发纸条

写留言

加关注



博客等级：**19**  
博客积分：**105**  
博客访问：**564,026**  
关注人气：**384**  
获赠金笔：**89**  
赠出金笔：**0**  
荣誉徽章：

相关博文

宋清辉：中国央行发降准“红包”  
宋清辉

超详细步骤教你制作家常美味—  
小衣服厨娘

市场人气有所恢复（19日收评）  
沙漠雄鹰

大盘将继续上涨布局此类龙头  
笑谭股市

谨防主力巧借魔咒发飙！  
南山听风

触底大反弹后主力将有大动作！  
笑谭股市

主力抄底三类股  
投资家老金

《红楼梦》中的美男人多是小鲜肉  
依涓涓

新的连板龙头诞生  
投资家老金

明天大盘收的越高越好  
趋势巡航

更多>>

正文

字体大小：大  中  小

## 操作系统 自旋锁+信号量+互斥量+临界区+死锁简介 (2016-06-12 21:28:56)

转  载  ▼

标签：  自旋锁  信号量  互斥量  临界区  死锁  分类：  操作系统服务器

### 自旋锁+信号量+互斥量+临界区+死锁简介

by:授客  QQ:  1033553122

#### 自旋锁（SpinLock）

自旋锁是专为防止多处理器并发而引入的一种锁。如果是单核处理器，则自旋锁定义为空操作，因为简单的关闭中断即可实现互斥。

自旋锁最多只能被一个线程持有，如果一个线程试图请求一个已被争用(已被另一个线程持有)的自旋锁，那么等待自旋锁的线程将会反复检查自旋锁是否释放，不会进入睡眠状态，一直处于忙等待状态(busy-waiting)，直到获取该自旋锁才能继续执行未完成的任务，所以常用于短期保护某段代码。

另外，持有自旋锁的进程也不允许睡眠，不然会造成死锁——因为睡眠可能造成持有锁的进程被重新调度，而再次申请自己已持有的锁。

事实上，自旋锁的初衷就是：在短期间内进行轻量级的锁定。一个被争用的自旋锁使得请求它的线程在等待锁重新可用的期间进行自旋(特别浪费处理器时间)，所以自旋锁不应该被持有时间过长。如果需要长时间锁定的话, 最好使用信号量。

自旋锁的基本形式如下：

```
spin_lock(&mr_lock);  
  
//临界区  
  
spin_unlock(&mr_lock);
```

同一时刻只有一个线程允许处于临界区中，可用来防止多处理器中并发访问临界区，抢占共享资源造成的竞争。

#### 信号量(Semaphore)

信号量是一种机制。在进入一个关键代码段之前，线程必须获取一个信号量；一旦执行完该关键代码段，则释放获取的信号量。如果获取不到信号量，则进入等待队列，并保持休眠状态(sleep-waiting)，此时，线程释放占用的处理器，以便处理器执行其它任务。当有未被持有的信号量时，唤醒队列中的线程，线程从而获取信号量，继续执行未完成的任务。

为了完成上述过程，需要创建一个信号量signal，然后将Acquire Semaphore signal以及Release Semaphore signal分别放置在每个关键代码段的首末端。确认这些信号量signal引用的是初始创建的信号量。

创建信号量时即要同时指出允许的最大资源计数和当前可用资源计数。一般是将当前可用资源计数设置为最大资源计数，每增加一个线程对共享资源的访问，当前可用资源计数就会减1，只要当前可用资源计数是大于0的，就可以发出信号量信号。但是当前可用计数减小到0时则说明当前占用资源的线程数已经达到了所允许的最大数目，不能在允许其他线程的进入，此时的信号量信号将无法发出。线程在处理完共享资源后，应在离开的同时将当前可用资源计数加1。在任何时候当前可用资源计数决不可能大于最大资源计数。

信号量基本使用形式为：

```
static DECLARE_MUTEX(mr_sem);//声明互斥信号量  
  
if(down_interruptible(&mr_sem))  
  
//可被中断的睡眠，当信号来到，睡眠的任务被唤醒  
  
//临界区  
  
up(&mr_sem);
```

信号量的睡眠特性，使得信号量可当作锁来用，特别是临界区需要被锁定较长时间(换句话说，需要较长时间持有锁)的情况；

注：信号量只能在进程上下文中使用，因为中断上下文中是不能被调度的；另外当线程持有信号量时，不可以再持有自旋锁(即针对同一代码段，不能同时使用自旋锁和信号量机制)。

外卖小哥雨中痛哭：人过三十，你

武书连2018中国最强大学终极

日本的“援助交际”到底是怎样产

爱吃香椿的你，一定别忽略了这

公积金原来可以当钱花！那些你不

周碧华：美国为何要对中国釜底抽

咱虽然命不好，但是想得多啊！

网约车杀熟，三年前就来了，只是

“震惊中外”和“震惊世界”

今日头条危局：流量、利润、资本

查看更多>>

谁看过这篇博文

暂无访客

### 互斥量（Mutex）

线程之间这种因共享资源而产生的相互制约关系。能够保证多个线程对同一共享资源的互斥访问。

同临界区有些类似，只有拥有互斥对象(即互斥量)的线程才具有访问共享资源的权限，由于互斥对象只有一个，因此就决定了任何情况下此共享资源都不会同时被多个线程所访问。当占据资源的线程在任务处理完任务后应释放占有的互斥对象，以便其他线程在获得该互斥对象后得以访问资源。

使用互斥不仅仅能够在同一应用程序不同线程中实现资源的安全共享，而且可以在不同应用程序的线程之间实现对资源的安全共享

互斥量与临界区的作用非常相似，但互斥量是可以命名的，也就是说它可以跨越进程使用。所以创建互斥量需要的资源更多，所以如果只为了在进程内部是用的话使用临界区会带来速度上的优势并能够减少资源占用量。因为互斥量是跨进程的互斥量一旦被创建，就可以通过名字打开它。

### 信号量与互斥量

竞争信号量与互斥量时需要进行进程睡眠和唤醒，代价较高，所以不适于短期代码保护，适用于保护较长的临界区

互斥量与信号量的区别？

(1) 互斥量用于线程的互斥，信号量用于线程的同步

这是互斥量和信号量的根本区别，也就是互斥和同步之间的区别

互斥：是指某一资源同时只允许一个访问者对其进行访问，具有唯一性和排它性。但互斥无法限制访问者对资源的访问顺序，即访问是无序的

同步：是指在互斥的基础上（大多数情况），通过其它机制实现访问者对资源的有序访问。在大多数情况下，同步已经实现了互斥，特别是所有写入资源的情况必定是互斥的。少数情况是指可以允许多个访问者同时访问资源

(2) 互斥量值只能为0/1，信号量值可以为非负整数

也就是说，一个互斥量只能用于一个资源的互斥访问，它不能实现多个资源的多线程互斥问题。信号量可以实现多个同类资源的多线程互斥和同步。当信号量为单值信号量是，也可以完成一个资源的互斥访问

(3) 互斥量的加锁和解锁必须由同一线程分别对应使用，信号量可以由一个线程释放，另一个线程得到

### 临界区（Critical Section）

保证在某一时刻只有一个线程能访问关键数据的简便办法。在任意时刻只允许一个线程对共享资源进行访问。如果有多个线程试图同时访问同一临界区，那么在有一个线程进入后其他所有试图访问此临界区的线程将被挂起，并一直持续到进入临界区的线程离开。临界区在被释放后，其他线程可以继续抢占，并以此达到用原子方式操作共享资源的目的。

保证在某一时刻只有一个线程能访问数据的简便办法。在任意时刻只允许一个线程对共享资源进行访问。如果有多个线程试图同时访问临界区，那么在有一个线程进入后，其他所有试图访问此临界区的线程将被挂起，并一直持续到进入临界区的线程离开。临界区在被释放后，其他线程可以继续抢占，并以此达到用原子方式操作共享资源的目的。

在使用临界区时，一般不允许其运行时间过长，只要进入临界区的线程还没有离开，其他所有试图进入此临界区的线程都会被挂起而进入到等待状态，并会在一定程度上影响。程序的运行性能。尤其需要注意的是不要将等待用户输入或是其他一些外界干预的操作包含到临界区。如果进入了临界区却一直没有释放，同样也会引起其他线程的长时间等待。虽然临界区同步速度很快，但却只能用来同步本进程内的线程，而不可用来同步多个进程中的线程。

### 死锁

假设有一个或多个内核任务和一个或多个资源，每个内核都在等待其中的一个资源，但所有的资源都已经被占用了，也就是说。这便会发生所有内核任务都在相互等待，但它们永远不会释放已经占有的资源，于是任何内核任务都无法获得所需要的资源，无法继续运行，这便意味着死锁发生了。自死琐是说自己占有了某个资源，然后自己又申请自己已占有的资源，显然不可能再获得该资源，因此就自缚手脚了。

0

喜欢

0

赠金笔

分享：

阅读(535) | 评论 (0) | 收藏(0) | 转载(0) | 喜欢▼ | 打印 | 举报

已投稿到： 排行榜

前一篇：Linux Linux内核参数调优

后一篇：Loadrunner 脚本开发-基于HTTP协议的流媒体视频在线播放服务器性能测试







评论

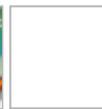



重要提示：警惕虚假中奖信息





[发评论]

做第一个评论者吧！[抢沙发>>](#)

发评论

[更多>>](#)





登录名： 密码： [找回密码](#) [注册](#) ☒ 记住登录状态

☐ 评论并转载此博文

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

[< 前一篇](#)

[Linux](#) [Linux内核参数调优](#)

[后一篇 >](#)

[Loadrunner](#) [脚本开发-基于HTTP协议的流媒体视频在线播放服务器性能测试](#)

新浪BLOG意见反馈留言板 不良信息反馈 电话：4006900000 提示音后按1键（按当地市话标准计费） 欢迎批评指正

[新浪简介](#) | [About Sina](#) | [广告服务](#) | [联系我们](#) | [招聘信息](#) | [网站律师](#) | [SINA English](#) | [会员注册](#) | [产品答疑](#)

Copyright © 1996 - 2018 SINA Corporation, All Rights Reserved

新浪公司 版权所有