

# Linux线程同步-----互斥量(Mutex)

原创

2016年07月22日 09:28:31

3348

2

## 互斥量

与信号处理函数一样，线程在访问全局资源时也会遇到非原子操作导致的冲突(可重入问题). 比如两个线程要对同一个寄存器加1，并行访问时可能会导致只加了一次.

CPU1执行 线程A的指令	CPU2执行 线程A的指令	变量的内存 单元的值
mov 0x8049540, %eax (eax = 5)	其它指令	5
add \$0x1, %eax (eax = 6)	mov 0x8049540, %eax (eax = 5)	5
mov %eax, 0x8049540 (eax = 6)	add \$0x1, %eax (eax = 6)	6
其它指令	mov %eax, 0x8049540 (eax = 6)	6

不可重入操作的特点时，输出不仅依赖于输入，还依赖于状态, 比如加1 依赖于状态，这个状态是寄存器原值. 访问状态和修改状态不是原子操作的话，就会导致并发冲突。

## 生成锁

Mutex用pthread\_mutex\_t类型的变量表示，可以这样初始化和销毁：

```
[cpp]
1. #include <pthread.h>
2. int pthread_mutex_destroy(pthread_mutex_t *mutex);
3. int pthread_mutex_init(pthread_mutex_t *restrict mutex, const pthread_mutexattr_t *restrict attr);
4. pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
```

成功返回0，失败返回错误号  
pthread\_mutex\_init 对Mutex 做初始化，参数attr 用于设定Mutex 属性。如果用PTHREAD\_MUTEX\_INITIALIZER 初始化全局或者静态Mutex, 那它相当于用pthread\_mutex\_init 初始化时attr 为NULL。

## 加锁与释锁

```
[cpp]
1. #include <pthread.h>
2. int pthread_mutex_lock(pthread_mutex_t *mutex);
3. int pthread_mutex_trylock(pthread_mutex_t *mutex);
4. int pthread_mutex_unlock(pthread_mutex_t *mutex);//成功返回0，失败返回错误号
```

一个线程调用pthread\_mutex\_lock 时，如果锁被其它线程占用，它将被挂起等待，直到另外一个线程调用pthread\_mutex\_unlock 释放Mutex, 当前线程被唤醒后试图重新加锁。只有加锁成功，当前线程才能继续执行。  
如果不想线程加锁时被挂起等待，可以用pthread\_mutex\_trylock。

## Mutex 锁的原理

假设Mutex 锁为1 时表示锁空闲，为0 表示锁占用。那么lock / unlock 的实现伪码如下：

```
[cpp]
1. lock:
2. if(mutex > 0)
3. {
4.     mutex = 0;
5.     return 0;
6. }
7. else
8. {
9.     //suspend;
10.    goto lock;
11. }
12. unlock:
13.     mutex = 1;
14.     唤醒其它因等待Mutex 而挂起的线程；
15. return 0;
```

可以看到，unlock 只有一个操作，可汇编码为单条指令，它是原子性的(单个cpu 时钟内，就能执行一条完成的汇编命令) 但是lock, 包含两个操作：一个是对mutex 的状态判断，一个是对mutex 状态设置。这两个操作, 有因果关系，必须合并为一个原子操作才能避免多个线程同时加锁成功。 其实，大多数汇编指令都提供了swap 或者 exchange 指令，这个指令就可以实现将: 状态判断和状态设置 合并为一个原子操作。以x86 的xchg 实现lock 和 unlock:

```
[cpp]
1. lock:
2.     movb $0, %a1;
3.     xchgb %a1, mutex;//如果mutex 为0（被占用状态）那么交换后a1 得到的是占用状态，原锁mutex 值不变（继续被占用）。
```

潇湘风凌

原创	粉丝	喜欢	评论
13	0	1	0

恒

等级： 博客 2 访问量：8500 积分：214 排名：34万+

- 博主最新文章 更多文章
- libevent学习笔记 ---- 回显服务器（3）
- libevent学习笔记 ---- 回显服务器（2）
- libevent学习笔记 ---- 回显服务器（1）
- Linux线程同步-----条件变量
- Linux线程同步-----读写锁

Linux	12篇
C	11篇
网络编程	2篇

2016年8月	3篇
2016年7月	3篇
2016年6月	1篇
2016年4月	1篇
2016年1月	5篇

- 博主热门文章
- Linux线程同步-----互斥量(Mutex) 3285
- C语言去除字符串首尾空格，trim()函数实现 2351
- Linux线程同步-----条件变量 431
- Linux下删除virtual vlan interface 406
- Linux下ifconfig的简易实现 357
- 判断IP、MAC地址是否符合规范 285
- Liunx下用代码设置ip地址 237
- Linux下通过shell批量替换文件内容 214
- Linux线程同步-----读写锁 209
- Linux下获取调用堆栈地址 201

```
7.         //suspend;
8.         goto lock;
9.     }
10. unlock:
11.     movb $1, mutex;
12.     唤醒其它因等待Mutex 而挂起的线程;
13.     return 0;
```



2

## Dead Lock 死锁

死锁有以下典型的情况： 1. 带锁线程的自己调用自己，如果线程自己调用自己就会因等待自己的锁释放而无限等待。这和递归下一样，递归是串行调用自己，根本不要锁机制。 2. 两个带锁线程AB 间的相互调用：如果线程A 获得锁lock1, 线程B 获得锁lock2. 此时线程A想获得lock2, 需要等待线程B 释放lock2, 但是线程B 又想获得锁lock1 B又等待A. 相当于AB 相互调用，相互等待对方释放锁。其实也相当于A 通过B 间接调用自己。经典的死锁问题: 5个哲学家就餐问题，就属于此类范畴(每位哲学都要等待右边的叉子释放)  
有几个方法避免发生死锁： 1. 方法一是按顺序加lock1, lock2,lock3 加锁lock3 之前，必须先加lock2, 加lock2 前，必须先加lock1. 释放lock 时必须按倒序来 2. 如果确定锁的顺序比较困难，则尽量用pthread\_mutex\_trylock 代替 pthread\_mutex\_lock 以避免死锁。 3. 用串行代替并行。

## Mutex小结

如果一个线程试图获取一个 mutex,但是没有成功,因为 mutex 已经被占用, 它将进入睡眠,让其他进程运行,直到 mutex 被其他进程释放,,两者都是开销比较大的操作,也就是 context switch 的开销.如果锁只是被其他线程占用非常短的时间,那么时间花在使的线程睡眠并唤醒它可能超过它使用 spinlock 持续获取锁的时间.

## Mutex使用示例

```
[cpp]
1.  #include <stdio.h>
2.  #include <stdlib.h>
3.  #include <string.h>
4.  #include <unistd.h>
5.  #include <pthread.h>
6.  #include <sys/types.h>
7.
8.  int seq = 0;
9.  pthread_mutex_t seq_lock = PTHREAD_MUTEX_INITIALIZER;
10.
11. int seq1 = 0;
12.
13. void *thr_func_without_lock(void *arg)
14. {
15.     printf("The sequence(without lock) is %d ...\r\n", ++seq1);
16. }
17.
18. void *thr_func_with_lock(void *arg)
19. {
20.     pthread_mutex_lock(&seq_lock);
21.     printf("The sequence is %d ...\r\n", ++seq);
22.     pthread_mutex_unlock(&seq_lock);
23.
24.     return ((void *)0);
25. }
26.
27. void demo()
28. {
29.     int i= 0;
30.     for (i = 0 ; i < 10000; ++i)
31.     {
32.         pthread_t tid1, tid2, tid3;
33.         pthread_create(&tid1, NULL, thr_func_without_lock, NULL);
34.         pthread_create(&tid2, NULL, thr_func_without_lock, NULL);
35.         pthread_create(&tid3, NULL, thr_func_without_lock, NULL);
36.         pthread_join(tid1, NULL);
37.         pthread_join(tid2, NULL);
38.         pthread_join(tid3, NULL);
39.     }
40. }
41.
42. void lock_demo()
43. {
44.     int i = 0;
45.     for (i = 0 ; i < 10000; ++i)
46.     {
47.         pthread_t tid1, tid2, tid3;
48.         pthread_create(&tid1, NULL, thr_func_with_lock, NULL);
49.         pthread_create(&tid2, NULL, thr_func_with_lock, NULL);
50.         pthread_create(&tid3, NULL, thr_func_with_lock, NULL);
51.         pthread_join(tid1, NULL);
52.         pthread_join(tid2, NULL);
53.         pthread_join(tid3, NULL);
54.     }
55. }
56.
57. int main(int argc, char **argv)
58. {
59.     demo();
60.     printf("\r\n");
61.     lock_demo();
62.     return 0;
63. }
```



✉ webmaster@csdn.net  
☎ 400-660-0108  
🗣 QQ客服 🗣 客服论坛

[关于](#) [招聘](#) [广告服务](#) [百度](#)

©1999-2018 CSDN版权所有  
京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/u013022032/article/details/51991610>



2



目前您尚未登录，请 [登录](#) 或 [注册](#) 后参与评论



## C++多线程-第二篇-Mutex(互斥量)



hffhjh111 2016-11-12 16:10:58 8869

//Boost #include #define BOOST\_THREAD\_VERSION 4 //使用最新版本，含有1,2,3但只是为了兼容之前程序。 Thread库丰富强大的扩展功能但不在Thre...

## Mutex（互斥锁）



longwang155069 2016-08-12 16:22:12 3492

互斥锁(mutex) 在信号量最后的部分说，当count=1的时候可以用信号量实现互斥。在早期的Linux版本中就是当count=1来实现mutex的。 在2.6.11版本中，如下：typed...

## 【C#拾遗】——Mutex对象深入理解



jiadajing267 2016-12-18 21:45:09 1702

最近小编新入职了一家公司，在熟悉系统的过程中发现Mutex对象，后来小编学习了一下，本文主要是讲解何为Mutex，以及Mutex的一些用法。 概念 C#中Mutex是互斥锁...

## 互斥锁mutex的使用方法



zgaoq 2017-02-08 14:56:18 3636

在线程实际运行过程中，我们经常需要多个线程保持同步。这时可以用互斥锁来完成任务；互斥锁的使用过程中，主要有pthread\_mutex\_init，pthread\_mutex\_destory，pthrea...

## 各种Mutex的使用与区别



zangchaodotcnatgmail 2016-03-10 15:33:21 1283

<http://blog.csdn.net/guosha/article/details/3136721> 版权声明：本文为博主原创文章，未经博主允许不得转载。 ...

## 互斥锁mutex



fhb1922702569 2016-12-30 21:01:30 581

在信号量最后的部分说，当count=1的时候可以用信号量实现互斥。在早期的Linux版本中就是当count=1来实现mutex的。 内核重新定义了一个新的数据结构 struct mutex, 将其称为...

## Event和Mutex区别



anjen 2009-10-26 09:33:00 5054

4月11日(原创)Event和Mutex区别事件事件是用来同步地位不相等的线程的，事件可以用来使一个线程完成一件事情，然后另外的线程完成剩下的事情。事件的使用很灵活，自动事件的激发态是由人工来控制的， ...

## 线程、同步与锁——Mutex想说爱你不容易



king16304 2016-08-10 16:48:24 790

除了Lock（）、Monitor之外，我们最长用的就是Mutex了，但是玩不好Mutex就总会造成死锁或者AbandonedMutexException（我就玩的不怎么好，在并发性访问测试的时候总是遇...

## C++使用Windows API CreateMutex函数多线程编程

C++中也可以使用Windows 系统中对应的API函数进行多线程编程。使用CreateThread函数创建线程，并且可以通过CreateMutex创建一个互斥量实现线程间数据的同步：#i...



dcrmg 2016-12-28 21:00:28 3014

## 线程同步之互斥量mutex的使用

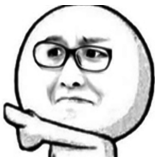


betsyfeng 2011-05-08 16:23:00 699

什么是Mutex “mutex” 是术语“互相排斥（mutually exclusive）”的简写形式，也就是互斥量。互斥量跟临界区中提到的Monitor很相似，只有拥有互斥对象的线程才具有访问资源的...

## 50万码农评论：英语对于程序员有多重要？

不背单词和语法，一个公式学好英语



//http://msdn.microsoft.com/zhcn/vcsharp/system.threading.mutex\_members.aspxMonitor通过向单个线程授予对象锁来控制对对...

Linux线程-互斥锁pthread\_mutex\_t

zmxiangde\_882012-09-20 07:31:4792726

在线程实际运行过程中，我们经常需要多个线程保持同步。这时可以用互斥锁来完成任务；互斥锁的使用过程中，主要有pthread\_mutex\_init，pthread\_mutex\_destory，pthrea...



2

mutex和condition的用法

wys72505782015-07-13 11:08:491477

分享一下DTCP-IP中用到的mutex和condition用法。Mutex一般有两种使用情况，一种是单独使用，保护临界区。一种是和condition配合使用，可以等待某condition拿到之后...

关于mutex的一些思考

occupy82015-08-22 22:00:52557

工作中遇到了死锁问题，先记录如下，欢迎大家提意见 先说说锁的定义：锁作为一种同步机制，是为了防止多个线程对临界资源的访问。这里请注意是仅仅只是临界资源。再来看看死...

函数 mutex\_init() / mutex\_lock() / mutex\_unlock()

1. 初始化互斥体 -- mutex\_init()；2. 获得互斥体 -- mutex\_lock()；3. 释放互斥体 -- mutex\_unlock()；mutex不能使用在 中断的上下...

jgw20082016-09-29 14:50:514037

如何正确释放Mutex

hello\_wyq2006-07-05 16:04:002117

在使用mutex的时候，我们往往会陷入因为忘记释放mutex而导致的麻烦，这个给程序员带来很多的麻烦，根据c++标准可知，局部变量在生命周期结束时，一定会被析构掉（除非在异常处理过程中，另外一个异常在...

操作系统如何实现mutex

GoOnDrift2014-02-10 17:37:525326

参考《linux c编程一站式学习》Mutex的两个基本操作lock和unlock是如何实现的呢？假设Mutex变量的值为1表示互斥锁空闲，这时某个进程调用lock可以获得锁，而Mutex的值为0...

关于Mutex的构造函数参数设置

JimFire2014-11-05 09:01:102922

Mutex的英文解释就是互斥体，也就是线程或者进程在同时访问一个资源的时候，是需要排他访问，也就是说一个时间段内只能有一个线程对象访问公共资源。Mutex的构造函数分为Mutex()，Mutex(...

线程（五）mutex

conganguo2014-10-26 20:53:36586

文章原始出处 http://xxinside.blogbus.com/logs/47162540.html 什么是Mutex “mutex” 是术语 “互相排斥 (mutually exclusiv...

Mutex::AutoLock介绍

southcamel2013-06-23 15:58:5415857

互斥类—Mutex Mutex是互斥类，用于多线程访问同一个资源的时候，保证一次只有一个线程能访问该资源。在《Windows核心编程》①一书中，对于这种互斥访问有一个很形象的比喻：想象你在飞机上如厕，...