问题1: 什么时候会释放Gil锁,

答 :

1 遇到像 i/o操作这种 会有时间空闲情况 造成cpu闲置的情况会释放Gil

2 会有一个专门ticks进行计数 一旦ticks数值达到100 这个时候释放Gil锁 线程之间开始竞争Gil锁(说明:

    ticks这个数值可以进行设置来延长或者缩减获得Gil锁的线程使用cpu的时间)

问题2: 互斥锁和Gil锁的关系

Gil锁  : 保证同一时刻只有一个线程能使用到cpu

互斥锁 : 多线程时,保证修改共享数据时有序的修改,不会产生数据修改混乱

首先假设只有一个进程,这个进程中有两个线程 Thread1,Thread2, 要修改共享的数据date, 并且有互斥锁

执行以下步骤

(1)多线程运行，假设Thread1获得GIL可以使用cpu，这时Thread1获得 互斥锁lock,Thread1可以改date数据(但并

没有开始修改数据)

(2)Thread1线程在修改date数据前发生了 i/o操作 或者 ticks计数满100 (注意就是没有运行到修改data数据),这个

时候 Thread1 让出了Gil,Gil锁可以被竞争

(3) Thread1 和 Thread2 开始竞争 Gil (注意:如果Thread1是因为 i/o 阻塞 让出的Gil Thread2必定拿到Gil,如果

Thread1是因为ticks计数满100让出Gil 这个时候 Thread1 和 Thread2 公平竞争)

(4)假设 Thread2正好获得了GIL, 运行代码去修改共享数据date,由于Thread1有互斥锁lock，所以Thread2无法更改共享数据

date,这时Thread2让出Gil锁 , GIL锁再次发生竞争

(5)假设Thread1又抢到GIL，由于其有互斥锁Lock所以其可以继续修改共享数据data,当Thread1修改完数据释放互斥锁lock,

Thread2在获得GIL与lock后才可对data进行修改

以上描述了 互斥锁和Gil锁的 一个关系

————————————————

版权声明：本文为CSDN博主「python胶水语言」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_41594007/article/details/79485847