**西安电子科技大学**

**操作系统课程设计**

**第2次实验报告**



实验题目： Priority-Scheduling--task1

学生姓名： 秦龙 (15030120044)

班 级： 1503012

指导教师： 吴锦桥

时 间： 2017年12月

**一、实验名称 Priority-Scheduling--task1**

**二、实验要求**

原始Pintos系统中对于线程的调度，没有考虑优先 级问题，采用的是最为简单的FCFS（先来先服务） 策略。

本实验要求为Pintos建立优先级调度机制，确保任 何时刻CPU上运行的都是最高优先级线程

**三、实验过程**

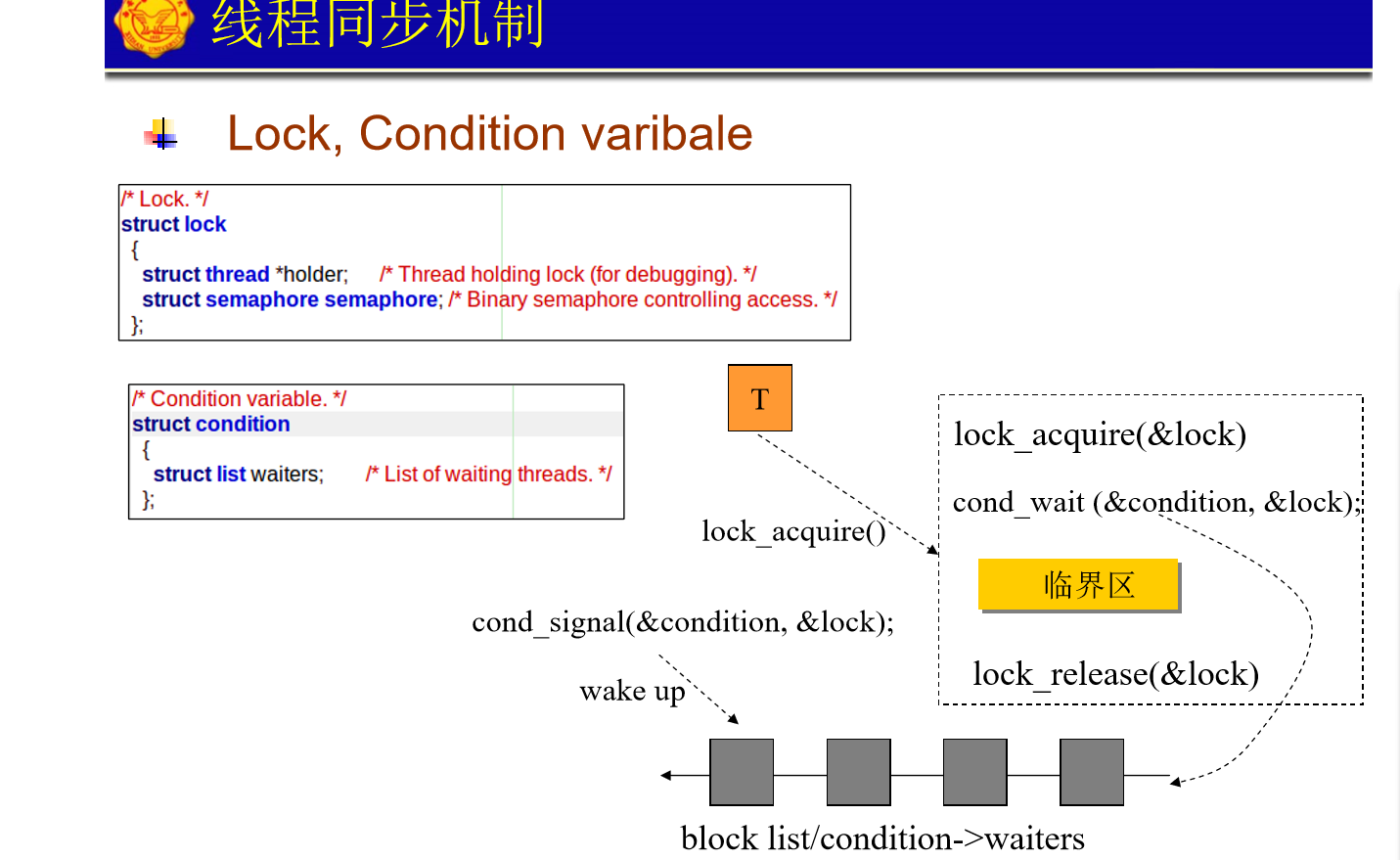
**1实验要求**

Pintos中的线程优先级如何定义；

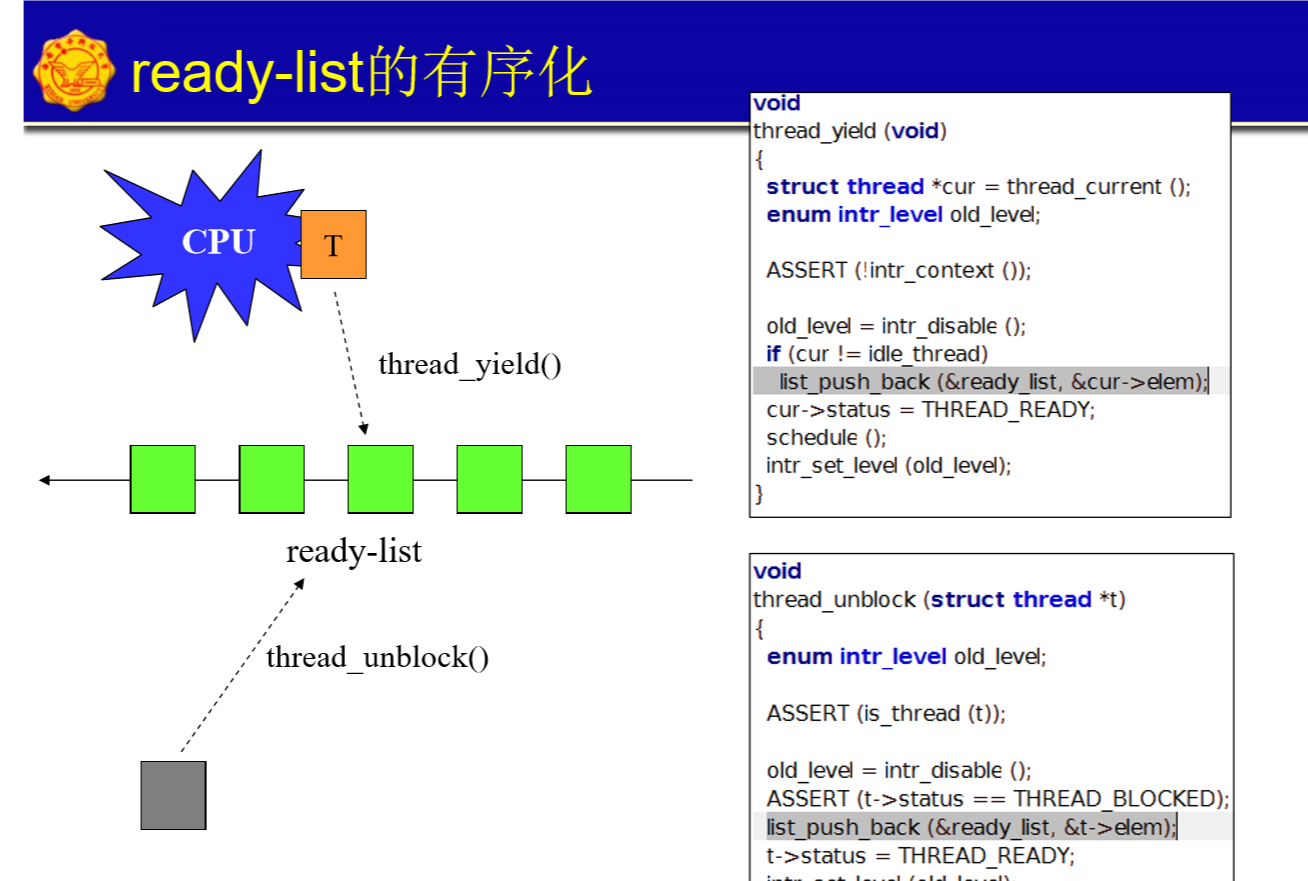
了解线程的优先级如何设置如何修改；

掌握导致优先级变化的原因；

本次试验中哪些情况下需要考虑优先级调度；



**上面是线程的同步机制**

****

**上面是线程的List调度队列**

**分析**

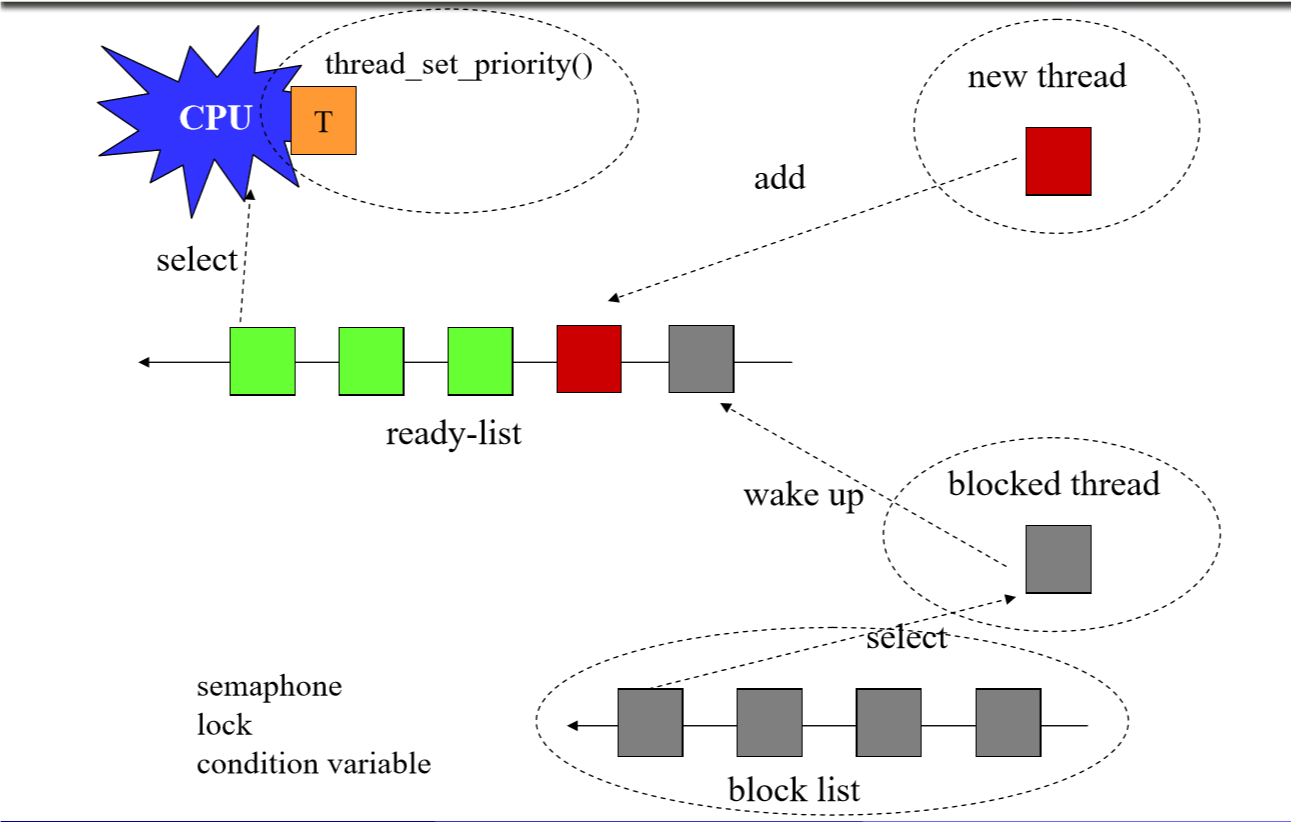
1分析可知，新建线程比当前的运行线程优先级高时，就会让当前线程让出CPU，这一过程使用函数thread\_yield()实现，但是该函数只是简单的把退出运行态的线程直接放到ready\_list队列中，因此为了达到进行优先级调度，我们必须要加入对优先级进行排序的函数。

2另外，thread\_unblock（）函数把blocked状态的线程标记为ready态，这里也同样需要对优先级进行排序，保证ready\_list中的线程始终保持按照优先级排序的有序状态。

3当一个正在运行的线程优先级发生改变时我们必须要判读改变后的优先级的大小，让其和ready\_list中的线程优先级进行比较，如果还是最高，则继续运行，否则让出CPU，进入ready\_list等待；

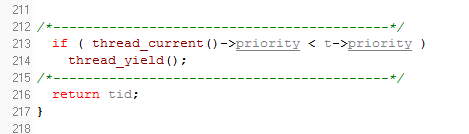
4如果多个线程因为等待一个信号量或者一个条件变量被阻塞，当条件满足时拥有最高优先级的线程应该优先被唤醒进入ready\_list，这里同样需要对其进行排序；

线程的调度会带来的一些影响

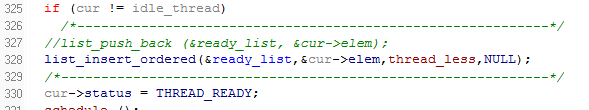


**设计过程**

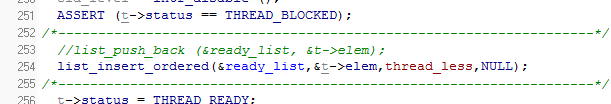
1在thread\_creat()函数中，如果新建的线程优先级大于CPU当前运行的线程，那么线程让出CPU，否则将线程有序放入ready\_list中



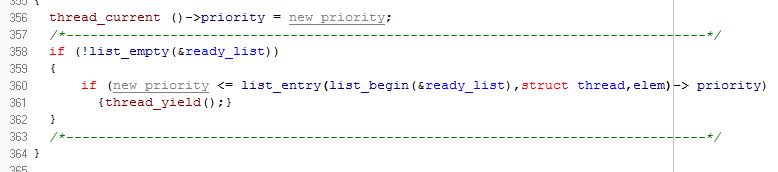
2Thread\_yield()函数中，当一个线程不满足优先级最高的条件得不到CPU调度进入ready\_list序列中，应该有序的插入而不是直接放到队尾或者其他的任意位置。因此我们需要对插入操作进行排序，此处我们使用系统自带的插入排序函数list\_insert\_ordered（）



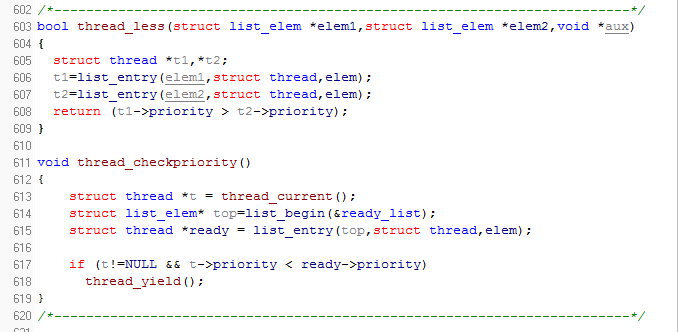
3Thread\_unblock()函数中，同理将一个线程从blocked态改变为ready态时，再插入ready\_list时同样需要对其进行排序操作；



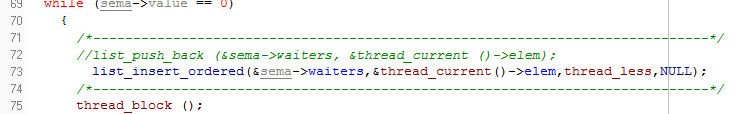
4Thread\_set\_priority()函数中，当一个函数的线程被改变时那么需要根据改变后的优先级进行判断，进行相应的比较，是线程去往该去的队列；



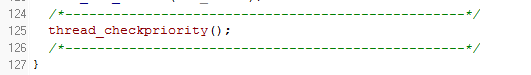
5为了达到比较优先级的效果我们需要自己添加一个优先级比较函数（需要在thread.h文件中声明）



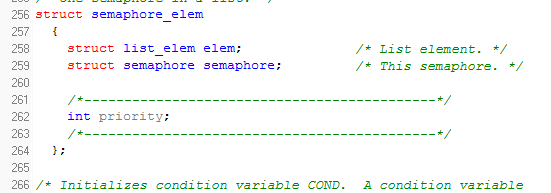
6在函数sema\_down（）中 由于一个信号量可以阻塞多个线程，需要在当条件满足时唤醒的总是最高优先级的线程，因此，当一个线程由于等待信号进入等待序列是，需要给阻塞线程的队列进行相同的排序：



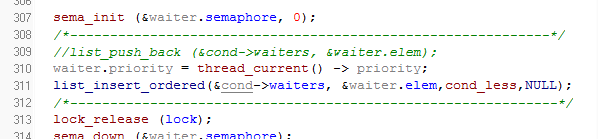
7在函数sema\_up（）中，同时添加相应的优先级检测函数调用：



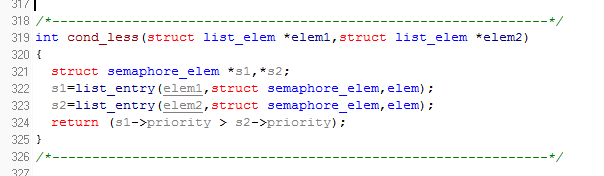
8由于信号量本身没有优先级这一概念，但是根据分析必须要根据类似于信号量的参数对阻塞的线程进行相应的排序，因此我们必须人为的添加一个模拟的信号量：



9在cond\_wait（）函数中，一个条件变量可能会被多个线程等待，但是我们必须要使得“优先级”高的线程得到条件变量满足运行的要求，所以我们要对等待条件变量的线程序列进行相应的排序保证每次条件变量分配给优先级最高的线程：



10同样这里我们需要定义一个关于阻塞队列的优先级比较函数（在synch.h头文件中声明）



**四、实验结果**



**五、心得体会**

这次的实验主要是在线程的优先级调度上面，通过调用相应的函数便可以实现对优先级的修改，通过优先级的设置便可以实现对多个优先级的相互协调调度。对系统的一些不足的地方做出了一些修改，对系统的认识也加深了。