操作系统 lab5 实验报告

计算机系 2019011312 姚建竹

实现的功能

1. 在ProcessControlBlockInner内加入三种数据结构available, allocation, need
2. 初始化PCB时要初始化该数据结构并push入0线程的vec
3. 实现check_deadlock算法，分别处理mutex和semaphore，使用指导书里的算法
4. create锁的时候设置available，为每个线程加入新的allocation和need
5. lock锁的时候如果available不为零，可以分配，availabel −= 1, allocation += 1；否则need+=1，如果需要检测死锁，检测并返回结果
6. unlock锁时如果队列为空，available += 1, allocation −=1，如果不为空，对应等待线程的need−=1, allocation+=1，原线程allocation−=1, available不变
7. enable_deadlock_detect：在pcb里设置一个bool变量表示是否启用
8. 创建线程时要加新的allocation, need

问答作业

在我们的多线程实现中，当主线程 (即 0 号线程) 退出时，视为整个进程退出， 此时需要结束该进程管理的所有线程并回收其资源。 - 需要回收的资源有哪些？ - 其他线程的 TaskControlBlock 可能在哪些位置被引用，分别是否需要回收，为什么？

1. 需要回收的资源有：当前process的所有tcb，recycle_res，所有子线程，地址空间memory_set，文件描述符表fd_table
2. task数组里面可能被引用，需要被回收。因为在memoryset释放前需要回收掉所有的tcb，否则会发生两次回收（见exit_current_and_run_next内注释），同时在地址空间的memory_set上也有引用。

对比以下两种 Mutex.unlock 的实现，二者有什么区别？这些区别可能会导致什么问题？

第一种无论是否队列中有线程，都会把锁置为false。
这是有问题的，如果把一个锁给了队列里的另一个线程，此时locked应为true，只有当队列中没有等待线程的时候，才可以把locked置为false
所以应该采用第二种Mutex2的写法。