HW6.

复习题 5.2 5.3 5.5 5.7 习题 5.4 5.6 5.12 5.13 5.14 5.15

5.2 产生并发的三种上下文是什么？

三种上下文为：多应用程序、结构化应用程序、操作系统结果

5.3 执行并发进程的最基本要求是什么？

加强互斥的能力.

5.5 竞争进程和合作进程有何区别？

竞争进程间没有信息交换，而合作进程间有共享数据。

竞争进程需要同时访问相同的资源，像磁盘、文件或打印机。

合作进程要么共享访问一个共有的资源，要么就与其它进程相互通信。

5.7 互斥的要求：

1. 必须强制实施互斥：在与相同资源或共享对象的临界区有关的所有进程中，一次只允许一个进程进入临界区。
2. 一个在非临界区停止的进程不能干涉其它进程。
3. 绝不允许出现需要访问临界区的进程被无限延迟的情况，即不会死锁或饥饿。

4. 没有进程在临界区中时，任何需要进入临界区的进程必须能够立即进入。
5. 对相关进程的执行速度和处理器的数量没有任何要求和限制。
6. 一个进程驻留在临界区中的时间必须是有限的。

习题.

5.4 a. $2 \le tally \le 100$

b. $2 \le tally \le 2 \times$ 进程数

5.6 考虑 turn=0 的情况：P(1) 会将 block[1] 置为 true，由于 block[0] 初始为 false，在 P(0) 过程中 block[0] 被置为 true，满足 (block[1-0])，turn 被置为 0，接着进入到临界区。接着 P(1) 中也满足 (turn != id)、(block[1-id])，turn 被置为 1，也进入到临界区。此时没有正确解决互斥问题。

5.12 两种定义效果相同。在前面的定义中，当信号量值为负时，我们能直观地知道当前等待的进程数量。这个程序中，无法显式地获知这个信息。但是两种定义可以相互替代。

5.13 a. 在第9行 semWait (block) 和第10行 semWait (mutex) 之间可能会有较长的时延，而在此期间其它进程可能变活跃而导致 must_wait 又一次变为 true。这样可能造成在临界区有4个进程使用资源。

b. 这会使未被阻塞的进程必须重新检查它们能否使用资源。但是会造成进程饥饿的问题。

5.14 a. 这个方法可以消除时延，因为一个进程在结束后会为它唤醒的进程更新 waiting 和 active 计数器，这些计数器的及时更新使得被释放的进程不必重新进入临界区。

b. 假设在资源紧张时有3个进程到达，但其中一个在被阻塞时错过了它的时间片。当最后一个进程结束时，它会执行3次 semSignal 操作并把 must_wait 置为 true。当在一个新的进程在之前的进程恢复执行前到达，则新进程会被阻塞。但是它会在未被阻塞时执行过 semWait，并且之前错过时间片的那个进程会在运行时被阻塞。

c. 执行结束后的进程会帮它唤醒出的进程更新系统状态，即替后一个完成。

5.15 a. 唤醒一个等待中的进程时，进程不会进入到互斥锁的状态而是自主地更新系统的状态和参数。结束后，它会重新打开互斥锁，因此新到达的进程不影响旧进程，因为在旧进程结束前无法进入互斥锁。

b. 上题的解法中只能唤醒一个等待中的进程，否则当前的状态变量会有冲突。本题的解法中能唤醒多个，因为，执行中新唤醒的进程会检查是否还有其它进程需要唤醒（若当前活跃的进程数<3），则唤醒另外的进程。

c. 每个进程结束后把进入活跃状态的许可交给下一个被阻塞的进程。这类似于有接力棒就能进入活跃状态。