作业7

1. 死锁产生的条件以及死锁解决的方案有哪些?

死锁产生的四个必要条件是：

1. **互斥条件**：一个资源每次只能被一个进程使用。
2. **占有和等待条件**：一个进程至少持有一个资源，并且正在等待获取其他进程持有的资源。
3. **不可抢占条件**：已经持有的资源不能被其他进程强行抢占，只能由持有它的进程自愿释放。
4. **循环等待条件**：存在一种进程资源的循环等待链，每个进程至少持有一个资源，并等待获取下一个进程所持有的资源。

解决死锁的方案主要有以下几种：

1. **预防死锁**：通过破坏死锁的四个必要条件之一来预防死锁的发生。例如，可以通过资源的静态预分配策略来破坏“占有和等待”条件，或者通过允许资源抢占来破坏“不可抢占”条件。
2. **避免死锁**：在资源的动态分配过程中，避免系统进入不安全状态。银行家算法是一种著名的避免死锁的算法，它通过谨慎地分配资源，确保系统始终处于安全状态。
3. **检测死锁**：允许系统进入可能的死锁状态，但通过检测来识别死锁是否发生，并采取措施解除死锁。常用的检测方法是资源分配图和银行家算法中的安全性算法。
4. **解除死锁**：当检测到死锁发生时，需要采取措施解除死锁。常用的解除死锁的方法包括：
    - **剥夺资源**：从某些进程中剥夺资源，分配给其他进程，以打破循环等待链。
    - **撤销进程**：终止一个或多个进程的执行，释放它们占有的资源，直到死锁环被打破。

2. 假设系统处于不安全状态，说明线程可以在不进入死锁状态的情况下完成执行。

系统处于不安全状态并不意味着系统一定会进入死锁状态。

- **进程在执行过程中，可能会提前终止**。当进程处于不安全状态的时候，因为OS当前资源紧缺或者进程执行过程发生异常，导致某些进程没有继续申请资源而被终止（被kill或异常终止），这样被终止的进程就会释放资源，让OS避开这次死锁。
- **进程在正常运行过程中，可能会提前释放部分资源。**
- **进程实际需要的最大资源小于声明的最大需求资源。**在安全性检查算法中，使用的数据结构是

需求矩阵Need和当前资源数Available，Need由最大需求矩阵Max减去已经分配的Allocation求得，Max是进程事先对自身所需资源的一个最坏情况下的预估（因为要满足运行，必定是>=实际需要的），但是在实际执行的情况中，可能进程实际上用不到这么多的资源，所以有可能就是这相差的资源数可以保证系统并非必然转换为死锁。

- **进程申请的资源为可消耗性资源**。对于可消耗性资源，是可以在进程运行过程中产生的（比如消息），因此对于某些阻塞的进程，虽然当前资源不足，但有可能在有限的时间内可以申请到得以运行完成的资源（可消耗性资源，由别的进程执行过程中产生），因此可以释放掉自己持有的资源，让其他进程因此也可以执行完毕。

3.

Consider the following snapshot of a system:

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
T_0	0 0 1 2	0 0 1 2	1 5 2 0
T_1	1 0 0 0	1 7 5 0	
T_2	1 3 5 4	2 3 5 6	
T_3	0 6 3 2	0 6 5 2	
T_4	0 0 1 4	0 6 5 6	

Answer the following questions using the banker's algorithm:

a. What is the content of the matrix ***Need***?

b. Is the system in a safe state?

c. If a request from thread T_1 arrives for (0,4,2,0), can the request be granted immediately?

a.

	ABCD
T0	0 0 0 0
T1	0 7 5 0
T2	1 0 0 2

T3	0 0 2 0
T4	0 6 4 2

b. Yes. T0, T2, T3, T4, T1

c. Yes.

	Allocation	Need	Available
	ABCD	ABCD	ABCD
T0	0 0 1 2	0 0 0 0	1 1 0 0
T1	1 4 2 0	0 3 3 0	
T2	1 3 5 4	1 0 0 2	
T3	0 6 3 2	0 0 2 0	
T4	0 0 1 4	0 6 4 2	

存在一个安全序列：T0, T2, T3, T4, T1

4. 如图所示的六个资源分配图中的哪一个死锁了？对于那些死锁的情况，提供线程和资源的等待 循环。对于没有死锁的情况，说明线程完成执行的顺序。

b, d发生了死锁。

a：T2，T3，T1

b：T1占有R1，等待R3；T3占有R3，等待R1

c：T2，T3，T1

d：T1，T2占有R1，等待R2；T3，T4占有R2，等待R1

e：T2，T1，T3，T4

f：T2，T4，T1，T3

5. 在内存管理系统中内存（部）碎片和外部碎片的区别是什么?

内部碎片是指内存分配时，分配的内存块大于实际需要的内存，导致剩余的未使用部分无法被有效利用。

外部碎片是指在内存中存在许多小的空闲内存块，这些块的总和足以满足某个分配请求，但由于它们不是连续的，无法进行分配。

区别：**内部碎片**与分配单元大小有关，是分配时因单元大于实际需求而导致的未使用空间。发生在已分配的内存块内部。**外部碎片**与内存的整体布局和分配策略有关，是因内存分配和释放导致的非连续空闲块。发生在内存的整体空闲区域。

6. 大多数系统允许程序在执行过程中为其地址空间分配更多内存。程序堆段中的数据分配就是这 样分配的内存的一个例子。在以下方案中，支持动态内存分配需要什么？ a.连续内存分配 b. 分页

a：在连续内存分配方案中，内存分配和管理要确保进程的地址空间是连续的。因此需要：

1. 内存紧缩。当内存中存在碎片时，内存紧缩可以将这些碎片压缩，形成足够大的连续空闲块来满足新的内存请求。
2. 适当的空闲块分配策略。
3. 空闲内存块管理，维护一个空闲内存块的列表。

b：分页方案中，内存分配不需要连续，内存被分成固定大小的页和页框，动态内存分配相对简单。为了支持动态内存分配，需要以下机制：

1. 页表
2. 页面置换算法
3. MMU将逻辑地址转换为物理地址

7. Consider a computer system with a 32-bit logical address and 8-KB page size. The system supports up to 1 GB of physical memory. How many entries are there in

each of the following? a. A conventional, single-level page table b. An inverted page table

a：

$2^{32} \div (8 \times 2^{10}) = 2^{19}$

b：

$2^{30} \div (8 \times 2^{10}) = 2^{17}$

8.

a：219+430=649

b：2300+10=2310

c：500 > 100, 地址无效

d：1327+400=1727

e：112 > 96, 地址无效