

1.

死锁产生的条件：

- 互斥：一次只有一个进程使用一个资源，进程不能访问分配给其他进程的资源
- 占有等待：一个进程等待其他进程持有资源时，继续占有已分配的资源
- 非抢占：进程不能强行抢走其他进程已经占有的资源
- 循环等待：存在闭合的进程链，每个进程占有链中下一个进程所需资源

解决方案：

- 死锁预防，破坏死锁产生条件
 - 破坏互斥条件：令可共享资源如可读文件不互斥
 - 破坏持有且等待：使进程在请求资源时，不持有任何资源。进程的资源持有情况只有全部持有和全部没有
 - 破坏无抢占：进程在持有资源的同时申请新资源：申请到新资源继续执行；申请不到新资源释放当前持有的所有资源并且进入资源等待队列
 - 破坏循环等待：抢占所有类型资源完全排序，进程以（资源的）递增序列请求资源
- 死锁避免，动态检查资源分配，确保不存在循环等待条件。系统评估根据请求分配资源后系统是否仍然处于安全状态，不处于安全状态的资源分配请求不执行
- 死锁检测和死锁恢复：允许系统进入死锁状态，但是定期检测系统中死锁的发生，并且取消所有的死锁进程进行重启，或者将每个死锁进程回滚到前面定义的某些检查点重新启动

2.

系统处于不安全状态指系统中不存在一个安全序列（进程执行顺序）使得所有进程能够不释放当前持有资源，完全没有死锁风险地完成执行。

系统处于不安全状态不意味着一定会有死锁发生：在不安全状态时可以令一些线程释放掉当前持有资源提前终止或者释放掉部分资源进入等待，让剩下的线程处于安全状态执行避开死锁风险，终止的线程可以看情况重新启动，等待的线程可以继续执行。

3.

	0	0	0	0
	0	7	5	0
a. Need	1	0	0	2
	0	0	2	0
	0	6	4	2

b. 存在安全序列：安全序列为 T0 T2 T3 T4 T1

c. 若资源分配给T0后T0能够立即执行结束，那么存在安全状态T0 T2 T3 T4 T1

4.

a. 无死锁，执行顺序 T2,T3,T1

b. 存在有向环，有死锁。线程T1等待线程T3释放资源R3，线程T3等待线程T1释放资源R1

c. 无死锁，执行顺序T2, T3, T1

d. 存在死锁有向环。线程T1,T2等待线程T3,T4释放资源R2，线程T3,T4等待线程T1,T2释放资源R1

e. 无死锁：T2,T1,T3,T4

f.无死锁: T2,T4,T1,T3

5.

内部碎片: 分配的内存大于请求的内存, 是已经分配的内存中未被利用的内存

外部碎片: 存在满足请求的总内存空间, 但是该空间不是连续的, 是未被分配的内存但是由于过小无法分配被利用

6.

a.

- 需要实现内存分区, 有固定分区, 可变分区等方案
- 需要实现内存保护, 保护用户进程彼此不受影响, 可以通过寄存器和MMU实现
- 需要列表维护已经分配的分区和空闲分区的信息

b.

- 需要页表将逻辑地址转换为物理地址
- 需要内存管理单元MMU将逻辑地址转换为物理地址
- 需要实现页面置换算法, 没有足够的空闲页框时需要页面置换

7.

一级页表: 一个页寻址需要 (字节粒度) 2^{13} , $32 - 13 = 19$, 2^{19} 个条目

倒排页表: 每个物理页一个条目 $2^{30} / 2^{13} = 2^{17}$, 2^{17} 个条目

8.

十六进制

a. $219 + 430 = 649$

b. $2300 + 10 = 2310$

c. 基地址为90, 但是偏移长度500超过了段的长度100, 地址无效

d. $1327 + 400$

e. 基地址1952, 但是偏移长度为112超过了段的长度96, 地址无效