**OS中期考试答案**

1. 计数信号量如何通过二元信号量和普通机器指令实现？

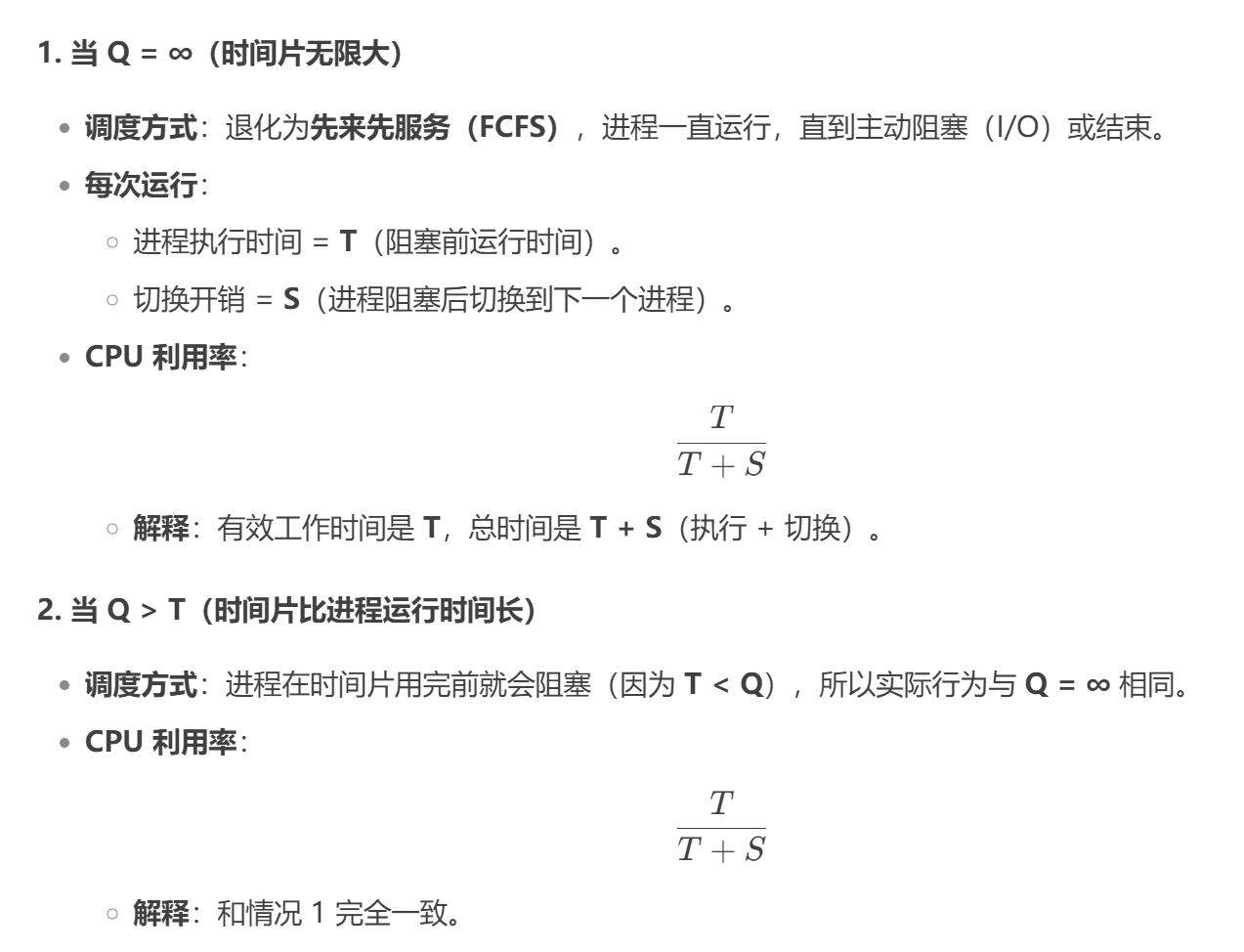
**答：**

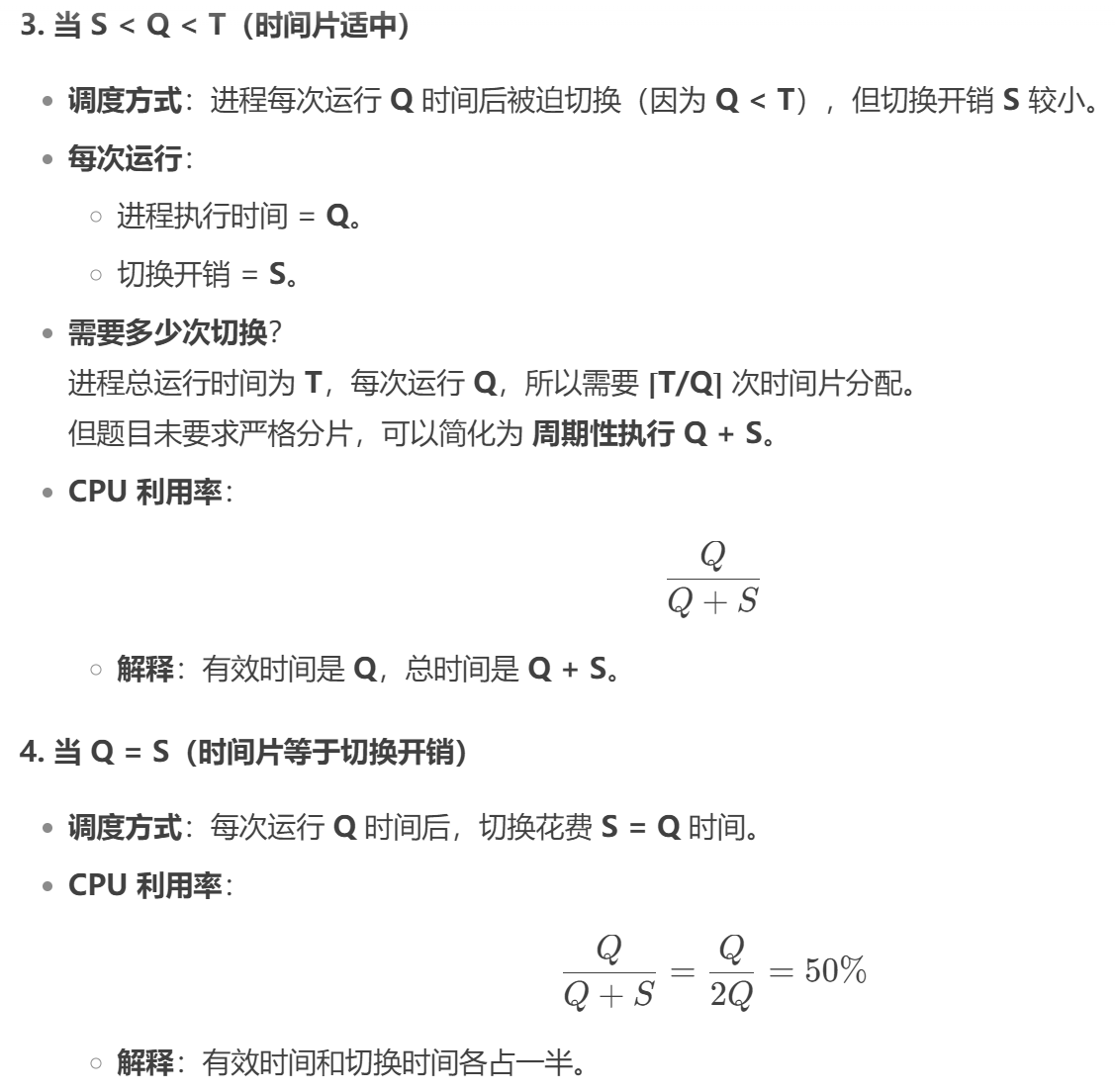
void semWait(semaphore s)  
{  
semWaitB(mutex);  
 s--;  
 if(s<0)  
 {  
semSignalB(mutex);  
semWaitB(delay);  
 }  
 else  
SemsignalB(mutex);  
 }  
 void semSignal(semaphore s);  
 {  
semWaitB(mutex);  
 s++;  
 if(s<=0)  
semSignalB(delay);  
semSignalB(mutex);  
 }

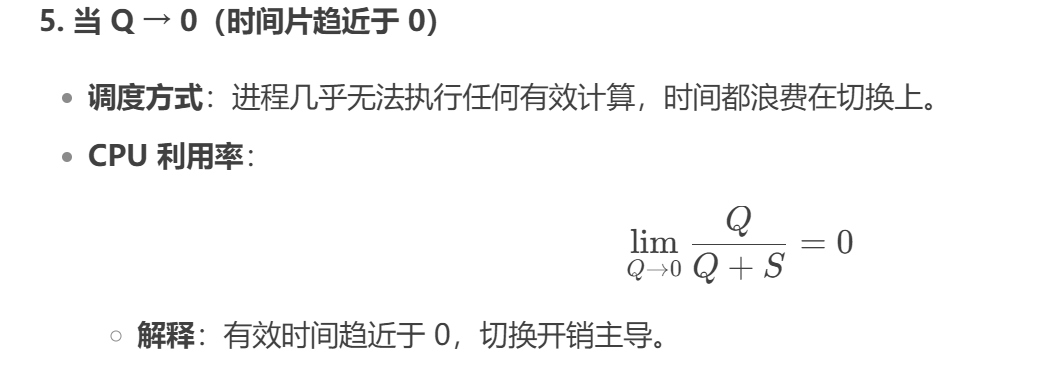
最初,S被设置成期待的信号量值,每个semWait操作将信号量减1,每个semSignal操作将信号量加1.二元信号量mutex被初始化成1,确保在更新s时保证互斥.二元信号量delay被初始化成0,用于挂起进程.

1. 监测表明：当进程阻塞在I/O之前时，平均每个进程运行时间为T。一次进程切换开销为S，对于采取时间片长度为Q的轮转调度，请给出以下各种情况中CPU的利用率
2. Q=∞, 2. Q>T, 3. S<Q<T, 4. Q=S, 5. Q趋于0

**答：**1. T/S+T, 2, T/S+T, 3. Q/S+Q, 4, 50%, 5. 0.







1. 在用户级空间实现线程的优点和缺点是什么？

**答：**在用户空间实现线程，其最大的优点是线程切换至少比陷入内核要快一个数量级；最大的缺点是程序员通常在经常发生线程阻塞的应用中才希望使用多个线程。

1. 系统中按内存地址排列的空闲区大小分别是：10KB, 4KB,20KB,18KB,7KB,9KB,12KB,15KB.对于连续的段请求：12KB,10KB,9KB。使用首次适应法，将找出哪个空闲区？使用最佳适应、最差适应、下次适应法呢？（画图）



答： 首次适应：

12KB 2# 10KB 0# 9KB 3#

最佳适应：

12KB 6# 10KB 0# 9KB 5#

最差适应：

12KB 2# 10KB 3# 9KB 7#

下次适应（循环首次适应）：

12KB 2# 10KB 3# 9KB 5#

1. 一个进程被分配了4个页框，页表如下。用NRU(即CLOCK算法), FIFO, LRU, 改进的时钟置换算法，分别将置换哪个页面？

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **页面** | **装入时间** | **上次访问时间** | **R** | **M** |
| 0 | 126 | 280 | 1 | 0 |
| 1 | 230 | 265 | 0 | 1 |
| 2 | 140 | 270 | 0 | 0 |
| 3 | 110 | 285 | 1 | 1 |

**答：** NRU: 1, FIFO:3, LRU, 1, 改进的时钟置换：2

1. 简单分页与虚拟内存分页有什么区别？

**答：**

简单分页：一个程序中的所有的页都必须在主存储器中程序才能正常运行，除非使用覆盖技术。

虚拟内存分页：不是程序的每一页都必须在主存储器的帧中来使程序运行，页在需要的时候进行读取。

1. 局部性原理是如何体现在虚拟内存技术中的？

**答：**局部性原理(principle of locality)：指程序在执行过程中的一个较短时期，所执行的指令地址和指令的操作数地址，分别局限于一定区域。还可以表现为：

时间局部性：一条指令的一次执行和下次执行，一个数据的一次访问和下次访问都集中在一个较短时期内（一条指令被执行了，则在不久的将来它可能再被执行）

空间局部性：当前指令和邻近的几条指令，当前访问的数据和邻近的数据都集中在一个较小区域内。（若某一存储单元被使用，则在一定时间内，与该存储单元相邻的单元可能被使用)。

局部性原理的具体体现：

程序在执行时，大部分是顺序执行的指令，少部分是转移和过程调用指令。

过程调用的嵌套深度一般不超过5，因此执行的范围不超过这组嵌套的过程。

程序中存在相当多的循环结构，它们由少量指令组成，而被多次执行。

程序中存在相当多对一定数据结构的操作，如数组操作，往往局限在较小范围内。

1. 什么是逻辑地址、物理地址、相对地址

**答：**

逻辑地址是指用户程序经编译后，每个目标模块以0为基地址进行的顺序编址。

物理地址是指内存中各物理存储单元的地址从统一的基地址进行的顺序编址。物理地址又称绝对地址，它是数据在内存中的实际存储地址。

相对地址是逻辑地址的一个特例，是相对于某些已知点（通常是程序的开始处）的存储单元。

1. 什么是内部碎片、外部碎片？

外部碎片：因为进程持续地被载入与换出，使得可用的内存空间被分割成许多不连续的区块。虽然内存所剩空间总和足够让新进程执行，却因为空间不连续，导致程序无法载入执行。外部碎片是在内存分段中在段的换入换出中产生的碎片。利用紧凑技术或分页可以消除外部碎片。

内部碎片：发生在以固定长度分区来进行分配的内存中。当一个程序载入到固定大小的分区时，假如程序小于分区，则剩余的空间将无法被使用，称为内部碎片。内部碎片是在内存分页中最后一页往往占不满所产生的碎片。

1. 页和帧（物理块）之间有什么区别？页和段有什么区别？

**答：页和帧区别：**

帧指把内存空间分成与页面相同大小的若干个存储块。

页是在内存分页存储管理中将一个进程的逻辑地址空间分成若干个大小相等的片,称为页面或页。

**页和段区别：**

1、页是信息的物理单位，分页是为了系统管理内存的方便而进行的，故对用户而言是不可见的，是透明的；段是信息的逻辑单位，分段是程序逻辑上的要求，对用户而言，分段是可见的。

2、页的大小固定，由系统决定；段的大小不是固定的，由用户程序本身决定。

3 、从用户角度看，页的地址空间是一维的，而段的地址空间是二维的。

1. 什么是用户模式、内核模式？

**答：**①内核模式

当CPU运行于内核模式时，一切程序都可运行。任务可以执行特权级指令，对任何I/O设备有全部的访问权，还能够访问任何虚地址和控制虚拟内存硬件。这种模式对应8086的ring0层，操作系统的核心部分，包括设备驱动程序都运行在该模式。

②用户模式

这个模式中，硬件防止特权指令的执行，并对内存和I/O空间的访问操作进行检查。这就允许WindowsNT4.0限制任务对各种I/O操作的访问，并捕捉违反系统完整性的任何行为。在用户模式中，运行的代码如果不通过操作系统中的某种机制，就不能进入内核模式。在8086处理器上，这个模式对应于 ring3层，操作系统的用户接口部分以及所有的用户应用程序都运行在该级别。

1. 什么是进程切换、模式切换？

**答：**进程切换：在某一时刻，一个正在运行的进程被中断，操作系统指定另一个进程为运行态，并把控制权交给这个进程。

模式切换：处理器在用户态和核心态之间的切换。

**进程切换、模式切换之间的区别：**

发生模式切换可以不改变正处于运行态的进程状态，在这种情况下，保存上下文环境和以后恢复上下文环境只需要很少的开销。但是，如果当前正在运行的进程被转换到另一个状态，则操作系统必须使其环境产生实质性的变化。

1. 死锁的四个必要条件？

**答：**

1. 互斥条件（Mutual exclusion）：资源不能被共享，只能由一个进程使用。
2. 请求与保持条件（Hold and wait）：已经得到资源的进程可以再次申请新的资源。
3. 非剥夺条件（No pre-emption）：已经分配的资源不能从相应的进程中被强制地剥夺。
4. 循环等待条件（Circular wait）：系统中若干进程组成环路，该环路中每个进程都在等待相邻进程正占用的资源。
5. 同步的准则？

**答：**所有的同步机制都应遵循以下四条准则：（1）空闲让进：因为，当无进程处于临界区时，表明临界资源处于空闲状态，应允许一个请求进入临界区的进程立即进入自己的临界区，以有效地利用临界资源。（2）忙则等待：因为，当已有进程进入临界区时，表明临界资源正在被访问，因而其它试图进入临界区的进程必须等待，以保证对临界资源的互斥访问。（3）有限等待：因为，对要求访问的临界资源的进程，应保证在有限时间内能进入自己的临界区，以免陷入“死等”状态。（4）让权等待：因为，当进程不能进入自己的临界区时，应立即释放处理机，以免进程陷入“忙等”状态