

# 《操作系统精髓与设计原理·第五版》练习题及答案

## 第1章计算机系统概述

1.1、图1.3中的理想机器还有两条I/O指令：

0011 = 从I/O中载入AC

0111 = 把AC保存到I/O中

在这种情况下，12位地址标识一个特殊的外部设备。请给出以下程序的执行过程（按照图1.4的格式）：

1.从设备5中载入AC。

2.加上存储器单元940的内容。

3.把AC保存到设备6中。

假设从设备5中取到的下一个值为3940单元中的值为2。

答案：存储器（16进制内容）：300：3005；301：5940；302：7006

步骤1：3005→IR；步骤2：3→AC

步骤3：5940→IR；步骤4：3+2=5→AC

步骤5：7006→IR；步骤6：AC→设备6

1.2、本章中用6步来描述图1.4中的程序执行情况，请使用MAR和MBR扩充这个描述。

答案：1. a. PC中包含第一条指令的地址300，该指令的内容被送入MAR中。

b. 地址为300的指令的内容（值为十六进制数1940）被送入MBR，并且PC增1。这两个步骤是并行完成的。

c. MBR中的值被送入指令寄存器IR中。

2. a. 指令寄存器IR中的地址部分（940）被送入MAR中。

b. 地址940中的值被送入MBR中。

c. MBR中的值被送入AC中。

3. a. PC中的值（301）被送入MAR中。

b. 地址为301的指令的内容（值为十六进制数5941）被送入MBR，并且PC增1。

c. MBR中的值被送入指令寄存器IR中。

4. a. 指令寄存器IR中的地址部分（941）被送入MAR中。

b. 地址941中的值被送入MBR中。

c. AC中以前的内容和地址为941的存储单元中的内容相加，结果保存到AC中。

5. a. PC中的值（302）被送入MAR中。

b. 地址为302的指令的内容（值为十六进制数2941）被送入MBR，并且PC增1。

c. MBR中的值被送入指令寄存器IR中。

6. a. 指令寄存器IR中的地址部分（941）被送入MAR中。

b. AC中的值被送入MBR中。

c. MBR中的值被存储到地址为941的存储单元之中。

1.4、假设有一个微处理器产生一个16位的地址（例如，假设程序计数器和地址寄存器都是16位）并且具有一个16位的数据总线。

a.如果连接到一个16位存储器上，处理器能够直接访问的最大存储器地址空间为多少？

b.如果连接到一个8位存储器上，处理器能够直接访问的最大存储器地址空间为多少？

c.处理访问一个独立的I/O空间需要哪些结构特征？

d.如果输入指令和输出指令可以表示8位I/O端口号，这个微处理器可以支持多少8位I/O端口？

答案：对于(a)和(b)两种情况，微处理器可以直接访问的最大存储器地址空间为 $2^{16} = 64K$  bytes；唯一的区别是8位存储器每次访问传输1个字节，而16位

存储器每次访问可以传输一个字节或者一个16位的字。对于(c)情况，特殊的输入和输出指令是必要的，这些指令的执行体会产生特殊的“I/O信号”

（有别于“存储器信号”，这些信号由存储器类型指令的执行体产生）；

在最小状态下，一个附加的输出引脚将用来传输新的信号。对于(d)情况，它支持 $2^8 = 256$ 个输入和 $2^8 = 256$ 个输出字节端口和相同数目的16位I/O端

口；在任一情况，一个输入和一个输出端口之间的区别是通过被执行的输

入输出指令所产生的不同信号来定义的。

1.5、考虑一个32位微处理器，它有一个16位外部数据总线，并由一个8MHz的输入时钟驱动。假设这个微处理器有一个总线周期，其最大持续时间等于4个输入时钟周期。请问该微处理器可以支持的最大数据传送速度为多少？外部数据总线增加到21位，或者外部时钟频率加倍，哪种措施可以更好地提高处理器性能？请叙述你的设想并解释原因。

答案：时钟周期 $= 1 / (8MHz) = 125ns$

总线周期 $= 4 \times 125ns = 500ns$

每500ns传输2比特；因此传输速度 $= 4MB/s$

加倍频率可能意味着采用了新的芯片制造技术（假设每个指令都有相同的时钟周期数）；加倍外部数据总线，在芯片数据总线驱动/锁存、总线控制逻辑的修改等方面手段广泛（或许更新）。在第一种方案中，内存芯片的速度要提高一倍（大约），而不能降低微处理器的速度；第二种方案中，内存的字长必须加倍，以便能发送/接受32位数量。

1.6、考虑一个计算机系统，它包含一个I/O模块，用以控制一台简单的键盘/打印机电传打字设备。CPU中包含下列寄存器，这些寄存器直接连接到系统总线上：INPR：输入寄存器，8位

OUTR：输出寄存器，8位

FGI：输入标记，1位

FGO：输出标记，1位

IEN：中断允许，1位

I/O模块控制从打字机中输入击键，并输出到打印机中去。打字机可以把一个字母数字符号编码成一个8位字，也可以把一个8位字解码成一个字母数字符号。当8位字从打字机进入输入寄存器时，输入标记被置位；当打印一个字时，输出标记被置位。

a.描述CPU如何使用这4个寄存器实现与打字机间的输入/输出。

b.描述通过使用IEN，如何提高执行效率？

答案：a.来源于打字机的输入储存在INPR中。只有当FGI=0时，INPR才会接收来自打字机的数据。当数据接收后，被储存在INPR里面，同时FGI置为1。

CPU定期检查FGI。如果FGI=1，CPU将把INPR里面的内容传送到AC，并把FGI置为0。

当CPU需要传送数据到打字机时，它会检查FGO。如果FGO=0，CPU处于等待。如果FGO=1，CPU将把AC的内容传送到OUTER并把FGO置为0。

当数字符号打印后，打字机将把FGI置为1。

b. (A) 描述的过程非常浪费。速度远高于打字机的CPU必须反复不断的检

查FGI和FGO。如果中断被使用，当打字机准备接收或者发送数据时，可以向CPU发出一个中断请求。IEN计数器可以由CPU设置（在程序员的控制下）。

1.7、实际上在所有包括DMA模块的系统中，DMA访问主存储器的优先级总是高于处理器访问主存储器的优先级。这是为什么？

答案：如果一个处理器在尝试着读或者写存储器时被挂起，通常除了一点轻微的时间损耗之外没有任何危害。但是，DMA可能从或者向设备(例如磁盘或磁带)以数据流的方式接收或者传输数据并且这是不能被打断的。否则，如果DMA设备被挂起（拒绝继续访问主存），数据可能会丢失。

1.9、一台计算机包括一个CPU和一台I/O设备D，通过一条共享总线连接到主存储器M，数据总线的宽度为1个字。CPU每秒最多可执行106条指令，平均每条指令需要5个机器周期，其中3个周期需要使用存储器总线。存储器读/写操作使用1个机器周期。假设CPU正在连续不断地执行后台程序，并且需要保证95%的指令执行速度，但没有任何I/O指令。假设1个处理器周期等于1个总线周期，现在要在M和D之间传送大块数据。

a.若使用程序控制I/O，I/O每传送1个字需要CPU执行两条指令。请估计通过D的I/O数据传送的最大可能速度。

b.如果使用DMA传送，请估计传送速度。

答案：a.处理器只能分配5%的时间给I/O.所以最大的I/O指令传送速度是 $10^6 \times 0.05 = 50000$ 条指令/秒。因此I/O的传送速率是25000字/秒。

b.使用DMA控制时，可用的机器周期下的数量是

$$10^6 (0.05 \times 5 + 0.95 \times 2) = 2.15 \times 10^6$$

如果我们假设DMA模块可以使用所有这些周期，并且忽略任何设置和状态检查时间，那么这个值就是最大的I/O传输速率。

1.10、考虑以下代码：

```
for (i = 0; i < 20; i++)
```

```
for (j = 0; j < 10; j++)
```

```
a[i] = a[i]*j
```

a.请举例说明代码中的空间局部性。

b.请举例说明代码中的时间局部性。

答案：a.读取第二条指令是紧跟着读取第一条指令的。

b.在很短的间歇时间内，a[i]在循环内部被访问了十次。

1.11、请将附录1A中的式（1.1）和式（1.2）推广到n级存储器层次中。

答案：定义：

$C_i$  = 存储器层次i上每一位的存储单元平均花销

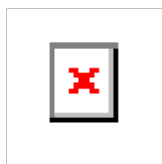
$S_i$  = 存储器层次i的规模大小

$T_i$  = 存储器层次i上访问一个字所需时间

$H_i$  = 一个字在不高于层次i的存储器上的概率

$B_i$  = 把一个数据块从层次i+1的存储器上传输到层次i的存储器上所需时间

高速缓冲存储器作为是存储器层次1；主存为存储器层次2；针对所有的N层存储器层以此类推。有：



T s的引用更复杂，我们从概率论入手：所期望的值，由此我们

可以写出：

我们需要清楚如果一个字在M1（缓存）中，那么对它的读取非常快。如果这个字在M2而不在M1中，那么数据块需要从M2传输到M1中，然后才能读取。因此， $T_2 = B_1 + T_1$ 进一步， $T_3 = B_2 + T_2 = B_1 + B_2 + T_1$

以此类推：

所以，

但是，

最后，

1.12、考虑一个存储器系统，它具有以下参数：

$T_c = 100 \text{ ns}$   $C_c = 0.01 \text{ 分/位}$

$T_m = 1200 \text{ ns}$   $C_m = 0.001 \text{ 分/位}$

a.1MB的主存储器价格为多少？

b.使用高速缓冲存储器技术，1MB的主存储器价格为多少？

c.如果有效存取时间比高速缓冲存储器存取时间多10%，命中率H为多少？

答案：a.价格 =  $C_m \times 8 \times 10^6 = 8 \times 10^3 C = \$ 80$

b.价格 =  $C_c \times 8 \times 10^6 = 8 \times 10^4 C = \$ 800$

c.由等式1.1知： $1.1 \times T_1 = T_1 + (1-H)T_2$

$(0.1)(100) = (1-H)(1200)$

$H = 1190/1200$

1.13、一台计算机包括包括高速缓冲存储器、主存储器和一个用做虚拟存储器的磁盘。如果要存取的字在高速缓冲存储器中，存取它需要20ns；如果该字在主存储器中而不在高速缓冲存储器中，把它载入高速缓冲存储器需要60ns（包括最初检查高速缓冲存储器的时间），然后再重新开始存取；如果该字不在主存储器中，从磁盘中取到内存需要12ms，接着复制到高速缓冲存储器中还需要60ns，再重新开始存取。高速缓冲存储器的命中率为0.9，主存储器的命中率为0.6，则该系统中存取一个字的平均存取时间是多少（单位为ns）？

答案：有三种情况需要考虑：

字所在的位置	概率	访问所需时间（ns）
在缓存中	0.9	20
不在缓存，在主存中	$(0.1)(0.6) = 0.06$	$60 + 20 = 80$
不在缓存也不在主存中	$(0.1)(0.4) = 0.04$	$12\text{ms} + 60 + 20 = 12,000,080$

480026 ns

1.14、假设处理器使用一个栈来管理过程调用和返回。请问可以取消程序计数器而用栈指针代替吗？

答案：如果栈只用于保存返回地址。或者如果栈也用于传递参数，这种方案只有当栈作为传递参数的控制单元而非机器指令时才成立。这两种情况下可以取消程序计数器而用栈指针代替。在后者情况中，处理器同时需要一个参数和指向栈顶部的程序计数器。

## 第2章操作系统概述

2.1假设我们有一台多道程序的计算机，每个作业有相同的特征。在一个计算周期T中，一个作业有一半时间花费在I/O上，另一半用于处理器的活动。每个作业一共运行N个周期。假设使用简单的循环法调度，并且I/O操作可以与处理器操作重叠。定义以下量：

?时间周期=完成任务的实际时间

?吞吐量=每个时间周期T内平均完成的作业数目

?处理器使用率=处理器活跃（不是处于等待）的时间的百分比

当周期T分别按下列方式分布时，对1个、2个和4个同时发生的作业，请计算这些量：

- a. 前一般用于I/O，后一半用于处理器。
- b. 前四分之一和后四分之一用于I/O，中间部分用于处理器。

答：（a）和（b）的答案相同。尽管处理器活动不能重叠，但I/O操作能。

一个作业时间周期=NT 处理器利用率=50 %

两个作业时间周期=NT 处理器利用率=100 %

四个作业时间周期=（2N-1）NT 处理器利用率=100 %

2.2 I/O限制的程序是指如果单独运行，则花费在等待I/O上的时间比使用处理器的时间要多的程序。处理器限制的程序则相反。假设短期调度算法偏爱那些在近期石油处理器时间较少的算法，请解释为什么这个算法偏爱I/O限制的程序，但是并不是永远不受理处理器限制程序所需的处理器时间？

受I/O限制的程序使用相对较少的处理器时间，因此更受算法的青睐。然而，受处理器限制的进程如果在足够长的时间内得不到处理器时间，同一算法将允许处理器去处理此进程，因为它最近没有使用过处理器。这样，一个处理器限制的进程不会永远得不到处理器。

2.3请对优化分时系统的调度策略和用于优化多道程序批处理系统的调度策略进行比较。

分时系统关注的是轮转时间，时间限制策略更有效是因为它给所有进程一个较短的处理时间。批处理系统关心的是吞吐量，更少的上下文转换和更多的进程处

理时间。因此，最小的上下文转换最高效。

2.4系统调用的目的是什么？如何实现与操作系统相关的的系统调用以及与双重模式（内核模式和用户模式）操作相关的系统调用？

系统调用被应用程序用来调用一个由操作系统提供的函数。通常情况下，系统调用最终转换成在内核模式下的系统程序。

2.5在IBM的主机操作系统OS/390中，内核中的一个重要模块是系统资源管理程序（System Resource Manager,SRM），他负责地址空间（进程）之间的资源分配。SRM是的OS/390在操作系统中具有特殊性，没有任何其他的主机操作系统，当然没有任何其他类型的操作系统可以比得上SRM所实现的功能。资源的概念包括处理器、实存和I/O通道，SRM累计处理器、I/O通道和各种重要数据结构的利用率，它的目标是基于性能监视和分析提供最优的性能，其安装设置了以后的各种性能目标作为SRM的指南，这会基于系统的利用率动态的修改安装和作业性能特点。SRM依次提供报告，允许受过训练的操作员改进配置和参数设置，以改善用户服务。现在关注SRM活动的一个实例。实存被划分为成千上万个大小相等的块，称为帧。每个帧可以保留一块称为页的虚存。SRM每秒大约接受20次控制，并在互相之间以及每个页面之间进行检查。如果页未被引用或被改变，计数器增1。一段时间后，SRM求这些数据的平均值，以确定系统中一个页面未曾被触及的平均秒数。这样做的目的是什么？SRM将采取什么动作？

操作系统可以查看这些数据已确定系统的负荷，通过减少加在系统上的活跃作业来保持较高的平均利用率。典型的平均时间应该是两分钟以上，这个平均时间看起来很长，其实并不长。

### 第3章进程描述和控制

3.1.给出操作系统进行进程管理时的五种主要活动，并简单描述为什么需要它们。

答：用户进程和系统进程创建及删除。系统中的进程可以为信息共享、运算加速、模块化和方便并发地执行。而并发执行需要进程的创建和删除机制。

当进程创建或者运行时分配给它需要的资源。当进程终止时，操作系统需要收回任何可以重新利用的资源。

进程的暂停和继续执行。在进程调度中，当进程在等待某些资源时，操作系统需要将它状态改变为等待或就绪状态。当所需要的资源可用时，操作系统需要将它状态变为运行态以使其继续执行。

提供进程的同步机制。合作的进程可能需要共享数据。对共享数据的并行访问可能会导致数据冲突。操作系统必须提供进程的同步机制以使合作进程有序地执行，从而保证数据的一致性。

提供进程的通信机制。操作系统下执行的进程既可以是独立进程也可以是合作进程。合作进程之间必须具有一定的方式进行通信。

提供进程的死锁解决机制。在多道程序环境中，多个进程可能会竞争有限的资源。如果发生死锁，所有的等待进程都将永远不能由等待状态再变为运行态，资源将被浪费，工作永远不能完成。

3.2.在[PINK89]中为进程定义了以下状态：执行（运行）态、活跃（就绪）态、

阻塞态和挂起态。当进程正在等待允许使用某一资源时，它处于阻塞态；当进程正在等待它已经获得的某种资源上的操作完成时，它处于挂起态。在许多操作系统中，这两种状态常常放在一起作为阻塞态，挂起态使用本章中给

出的定义。请比较这两组定义的优点。

答：[PINK89]中引用了以下例子来阐述其中阻塞和挂起的定义：

假设一个进程已经执行了一段时间，它需要一个额外的磁带设备来写出一个临时文件。在它开始写磁带之前，进程必须得到使用某一设备的许可。

当它做出请求时，磁带设备可能并不可用，这种情况下，该进程就处于阻塞态。假设操作系统在某一时刻将磁带设备分配给了该进程，这时进程就重新变为活跃态。当进程重新变为执行态时要对新获得的磁带设备进行写操作。这时进程变为挂起态，等待该磁带上当前所进行的写操作完成。

这种对等待某一设备的两种不同原因的区别，在操作系统组织其工作时是非常有用的。然而这并不能表明那些进程是换入的，那些进程是换出的。后一种区别是必需的，而且应该在进程状态中以某种形式表现出来。

3.3.对于图3.9（b）中给出的7状态进程模型，请仿照图3.8（b）画出它的排队图。

答：图9.3给出了单个阻塞队列的结果。该图可以很容易的推广到多个阻塞队列的情形。

3.4.考虑图3.9（b）中的状态转换图。假设操作系统正在分派进程，有进程处于

就绪态和就绪/挂起态，并且至少有一个处于就绪/挂起态的进程比处于就绪态的所有进程的优先级都高。有两种极端的策略：

（1）总是分派一个处于就绪态的进程，以减少交换；（2）总是把机会给具有最高优先级的进程，即使会导致在不需要交换时进行交换。请给出一种能均衡考虑优先级和性能的中间策略。

答：对于一个就绪/挂起态的进程，降低一定数量（如一或两个）优先级，从而保证只有当一个就绪/挂起态的进程比就绪态的进程的最高优先级还高出几个优先级时，它才会被选做下一个执行。

3.5.表3.13给出了VAX/VMS操作系统的进程状态。

a.请给出这么多种等待状态的理由。

b.为什么以下状态没有驻留和换出方案：页错误等待、也冲突等待、公共事件等待、自由页等待和资源等待。

c.请画出状态转换图，并指出引发状态转换的原因。

答：

a.每一种等待状态都有一个单独的队列与其相关联。当影响某一等待进程的事件发生时，把等待进程分成不同的队列就减少了定位这一等待进程所需的工作量。例如，当一个页错误完成时，调度程序就可以在页错误等待队列中找到等待的进程。

b.在这些状态下，允许进程被换出只会使效率更低。例如，当发生页错误等待时，进程正在等待换入一个页从而使其可以执行，这是将进程换出是毫无意义的。

c.可以由下面的进程状态转换表得到状态转换图。

	当前正在 执行	可 计 算 (驻留)	可 计 算 (换出)	各种等待 状态 (驻 留)	各种等待 状态 (换 出)
当前正在 执行		重调度		等待	
可 计 算 (驻留)	调度		换出		
可 计 算 (换出)		换入			
各种等待 状态 (驻 留)		事件发生			换出
各种等待 状态 (换 出)			事件发生		

保护和共享。访问模式确定：

- 指令执行特权：处理器将执行什么指令。
- 内存访问特权：当前指令可能访问虚拟内存中的哪个单元。

四种模式如下：

- 内核模式：执行VMS操作系统的内核，包括内存管理、中断处理和I/O操作。
- 执行模式：执行许多操作系统服务调用，包括文件（磁盘和磁带）和记录管理例程。
- 管理模式：执行其他操作系统服务，如响应用户命令。
- 用户模式：执行用户程序和诸如编译器、编辑器、链接程序、调试器之类的实用程序。

在较少特权模式执行的进程通常需要调用在较多特权模式下执行的过程，例如，一个用户程序需要一个操作系统服务。这个调用通过使用一个改变模式（简称CHM）指令来实现，该指令将引发一个中断，把控制转交给处于新的访问模式下的例程，并通过执行REI（Return from Exception or Interrupt，从异常或中断返回）指令返回。

a.很多操作系统有两种模式，内核和用户，那么提供四种模式有什么优点和缺点？

b.你可以举出一种有四种以上模式的情况吗？

答：

a.四种模式的优点是对主存的访问控制更加灵活，能够为主存提供更好的保护。缺点是复杂和处理的开销过大。例如，程序在每一种执行模式下都要有一个独立的堆栈。

b.原则上，模式越多越灵活，但是四种以上的模式似乎很难实现。

3.7.在前面习题中讨论的VMS方案常常称为环状保护结构，如图3.18所示。3.3

节所描述的简单的内核/用户方案是一种两环结构，[SILB04]指出了这种方法的问题：环状（层次）结构的主要缺点是它不允许我们实施须知原理，特别地，如果一个对象必须在域D<sub>j</sub>中可访问，但在域D<sub>i</sub>中不可访问，则必须有就

j

a.请清楚地解释上面引文中提出的问题。

b.请给出环状结构操作系统解决这个问题的一种方法。

答:

a.当j

包含的信息比D i中的更具有特权或者要求的安全性更高, 那么这种限制就是合理的。然而, 通过以下方法却可以绕过这种安全策略。一个运行在D j中的进程可以读取D j中的数据, 然后把数据复制到D i中。随后, D i中的进程就可以访问这些信息了。

b.有一种解决这一问题的方法叫做可信系统, 我们将在16章中进行讨论。

3.8.图3.7 (b) 表明一个进程每次只能在一个事件队列中。

a.是否能够允许进程同时等待一个或多个事件? 请举例说明。

b.在这种情况下, 如何修改图中的排队结构以支持这个新特点?

答:

a.一个进程可能正在处理从另一个进程收到的数据并将结果保存到磁盘上。

如果当前在另一个进程中正有数据在等待被取走, 进程就可以继续获得数据并处理它。如果前一个写磁盘操作已经完成, 并且有处理好的数据在等待写出, 那么进程就可以继续写磁盘。这样就可能存在某一时刻, 进程即在等待从输入进程获得数据, 又在等待磁盘可用。

b.有很多种方法解决这一问题。可以使用一种特殊的队列, 或者将进程放入

两个独立的队列中。不论采用哪种方法, 操作系统都必须处理好细节工作, 使进程相继地关注两个事件的发生。

3.9.在很多早期计算机中, 中断导致寄存器值被保存在与给定的中断信息相关联

的固定单元。在什么情况下这是一种实用的技术? 请解释为什么它通常是不方便的。

答: 这种技术是基于被中断的进程A在中断响应之后继续执行的假设的。但是, 在通常情况下, 中断可能会导致另一个进程B抢占了进程A。这是就必须将进程A的执行状态从与中断相关的位置复制到与A相关的进程描述中。然而机器却有可能仍将它们保存到前一位置。参考: [BRIN73]。

3.10.3.4节曾经讲述过, 由于在内核模式下执行的进程是不能被抢占的, 因此

UNIX不适用于实时应用。请阐述原因。

答: 由于存在进程不能被抢占的情况 (如在内核模式下执行的进程), 操作系统不可能对实时需求给予迅速的反应。

#### 第4章线程、对称多处理和微内核

4.1.一个进程中的多个线程有以下两个优点: (1) 在一个已有进程中创建一个新

线程比创建一个新进程所需的工作量少; (2) 在同一个进程中的线程间的通信比较简单。请问同一个进程中的两个线程间的模式切换与不同进程中的两个线程间的模式切换相比, 所需的工作量是否要少?

答: 是的, 因为两个进程间的模式切换要储存更多的状态信息。

4.2.在比较用户级线程和内核级线程时曾指出用户级线程的一个缺点, 即当一个

用户级线程执行系统调用时, 不仅这个线程被阻塞, 而且进程中的所有线程

都被阻塞。请问这是为什么?



答：因为对于用户级线程来说，一个进程的线程结构对操作系统是不可见的，而操作系统的调度是以进程为单位的。

4.3.在OS/2中，其他操作系统中通用的进程概念被分成了三个独立类型的实体：

会话、进程和线程。一个会话是一组与用户接口（键盘、显示器、鼠标）相关联的一个或多个进程。会话代表了一个交互式的用户应用程序，如字处理程序或电子表格，这个概念使得PC用户可以打开一个以上的应用程序，在屏幕上显示一个或更多个窗口。操作系统必须知道哪个窗口，即哪个会话是活跃的，从而把键盘和鼠标的输入传递个相应的会话。在任何时刻，只有一个会话在前台模式，其他的会话都在后台模式，键盘和鼠标的所有输入都发送给前台会话的一个进程。当一个会话在前台模式时，执行视频输出的进程直接把它发送到硬件视频缓冲区。当一个会话在后台时，如果该会话的任何一个进程的任何一个线程正在执行并产生屏幕输出，则这个输出被送到逻辑视频缓冲区；当这个会话返回前台时，屏幕被更新，为新的前台会话反映出逻辑视频缓冲区中的当前内容。

有一种方法可以把OS/2中与进程相关的概念的数目从3个减少到2个。删去会话，把用户接口（键盘、显示器、鼠标）和进程关联起来。这样，在某一时刻，只有一个进程处于前台模式。为了进一步地进行构造，进程可以被划分成线程。

a.使用这种方法会丧失什么优点？

b.如果继续使用这种修改方法，应该在哪里分配资源（存储器、文件等）：在

进程级还是线程级？

答：

a.会话的使用非常适合个人计算机和 workstation 对交互式图形接口的需求。它为

明确图形输出和键盘/鼠标输入应该被关联到什么位置提供了一个统一的

机制，减轻了操作系统的工作负担。

b.应该和其他的进程/线程系统一样，在进程级分配地址空间和文件。

4.4.考虑这样一个环境，用户级线程和内核级线程呈一对一的映射关系，并且允

许进程中的一个或多个线程产生会引发阻塞的系统调用，而其他线程可以继续运行。解释为什么这个模型可以使多线程程序比在单处理器机器上的相应的单线程程序运行速度更快？

答：问题在于机器会花费相当多的时间等待I/O操作的完成。在一个多线程程序中，可能一个内核级线程会产生引发阻塞的系统调用，而其他内核级线程可以继续执行。而在单处理器机器上，进程则必须阻塞知道所有的系统调用都可以继续运行。参考：[LEWI96]

4.5.如果一个进程退出时，该进程的某些线程仍在运行，请问他们会继续运行吗？

答：不会。当一个进程退出时，会带走它的所有东西——内核级线程，进程结构，存储空间——包括线程。参考：[LEWI96]

4.6.OS/390主机操作系统围绕着地址空间和任务的概念构造。粗略说来，一个地

址空间对应于一个应用程序，并且或多或少地对应于其他操作系统中的一个进程；在一个地址空间中，可以产生一组任务，并且它们可以并发执行，这大致对应于多线程的概念。管理任务结构有两个主要的数据结构。地址空间

控制块（ASCB）含有OS/390所需要的关于一个地址空间的信息，而不论该地址空间是否在执行。ASCB中的信息包括分派优先级、分配给该地址空间的实存和虚存、该地址空间中就绪的任务数以及是否每个都被换出。一个任务控制块（TCB）标识一个正在执行的用户程序，它含有在一个地址空间中管理该任务所需要的信息，包括处理器状态信息、指向该任务所涉及到的程序的指针和任务执行结构。ASCB是在系统存储器中保存的全局结构，而TCB是保存在各自的地址空间中的局部结构。请问把控制信息划分成全局和局部两部分有什么好处？

答：关于一个地址空间的尽可能多的信息可以随地址空间被换出，从而节约了主存。

4.7.一个多处理系统有8个处理器和20个附加磁带设备。现在有大量的作业提交

给该系统，完成每个作业最多需要4个磁带设备。假设每个作业开始运行时只需要3个磁带设备，并且在很长时间内都只需要这3个设备，而只是在最后很短的一段时间内需要第4个设备以完成操作。同时还假设这类作业源源不断。

a.假设操作系统中的调度器只有当4个磁带设备都可用时才开始一个作业。

当作业开始时，4个设备立即被分配给它，并且直到作业完成时才被释放。

请问一次最多可以同时执行几个作业？采用这种策略，最多有几个磁带设

备可能是空闲的？最少有几个？

b.给出另外一种策略，要求其可以提高磁带设备的利用率，并且同时可以避免系统死锁。分析最多可以有几个作业同时执行，可能出现的空闲设备的范围是多少。

答：

a.采用一个保守的策略，一次最多同时执行 $20/4=5$ 个作业。由于分配各个任务的磁带设备最多同时只有一个空闲，所以在同一时刻最多有5个磁带设备可能是空闲的。在最好的情况下没有磁带设备空闲。

b.为了更好的利用磁设备，每个作业在最初只分配三个磁带设备。第四个只有需要的时候才分配。在这种策略中，最多可以有 $20/3=6$ 个作业同时执行。最少的空闲设备数量为0，最多有2个。参考：Advanced Computer Architecture, K. Hwang, 1993.

4.8.在描述Solaris用户级线程状态时，曾表明一个用户级线程可能让位于具有

相同优先级的另一个线程。请问，如果有一个可运行的、具有更高优先级的线程，让位函数是否还会导致让位于具有相同优先级或更高优先级的线程？

答：任何一个可能改变线程优先级或者使更高优先级的线程可运行的调用都会引起调度，它会依次抢占低优先级的活跃线程。所以，永远都不会存在一个可运行的、具有更高优先级的线程。参考：[LEVI96]

第5章并发性：互斥和同步

## 5.1

答：b.协同程序read读卡片，将字符赋给一个只有一个字大小的缓冲区rs然后在赋给squash协同程。协同程序Read在每副卡片图像的后面插入一个额外的空白。协同程序squash不需要知道任何关于输入的八十个字符的结构，它简单的查