

习题 1 操作系统概述

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	B	D	C	D	C	D	A	C	D	D
题号	11	12	13	14	15	16	17	18	19	20
答案	B	C	C	C	B	D	B	B	B	A

二、综合题

1、答：

并发性和并行性是既相似又有区别的两个概念。并行性是指两个或多个事件在同一时刻发生；而并发性是指两个或多个事件在同一时间间隔内发生。

在单处理器系统中只有一条指令流水线，一个多功能的操作部件，某时刻处理机只能执行一个进程，进程与进程之间不能并行执行，只能并发执行。但在各种 I/O 控制技术的帮助下，处理机、通道和设备之间都能进行并发。

- (1) 处理机和设备之间的并行，能够发生。
- (2) 处理机和通道之间的并行，能够发生。
- (3) 通道和通道之间的并行，能够发生。
- (4) 设备和设备之间的并行，能够发生。

2、答：

以多道程序设计技术为基础的现代操作系统具有 4 个基本特征：

- (1) 并发性：多个程序并发执行，宏观并行，微观串行。
- (2) 共享性：多个程序共享系统中的所有资源
- (3) 虚拟性：操作系统为每个进程都虚拟出了一整套其所需的软硬件资源，让进程所属的用户感觉自己独占整个系统。操作系统通过进程状态转换实现虚拟性。当进程被切换出去运行态时，它的运行环境被操作系统保存，当把再次被调度程序选中切换到运行态时恢复其运行环境继续上次运行状态继续运行。
- (4) 异步性：并发执行的各个进程之间运行时间、运行顺序具有不确定性，即异步性，程序执行已经失去的封闭性和可再现性。操作系统通过同步机制保证多个进程能够正确的执行。

3、答：

多道程序设计技术是指同时把多个程序放入内存并允许交替执行和共享系统中的各类资源，当一个程序因某种原因（如 I/O 请求）而暂停执行时，CPU 立即转去执行另一个程序。操作系统在引入多道程序设计技术后，使得系统内有了多个程序（进程），它们宏观上看同时执行，微观上看仍然是串行。

多道程序设计技术的优点：多道程序交替穿插执行，提高了 CPU、内存和 I/O 设备的利用率；在保持 CPU、I/O 设备不断工作的同时，导致系统吞吐量的上升。

4、答：

推动批处理系统形成和发展的主要动力是“不断提高系统资源利用率”和“提高系统吞吐量”。这主要表现在：脱机输入/输出技术的应用和作业的自动过渡大大地提高了 I/O 的速

度及 I/O 设备与 CPU 并行工作的程度,减少了主机 CPU 的空闲时间;多道程序设计技术的应用更进一步提高了 CPU、内存和 I/O 设备的利用率及系统的吞吐率。

推动分时系统形成和发展的主要动力是“为了更好地满足用户的需要”。这主要表现在:CPU 的分时使用缩短了作业的平均周转时间;人机交互能力的提高使用户能方便地直接控制自己的作业;主机的共享使多个用户(包括远程用户)能同时使用同一台计算机独立地、互补干扰地处理自己的作业。

5、答:

不确定性指在多道程序设计环境下,系统中每道程序的推进时间、顺序以及完成时间由于受其运行环境的影响是不确定的、不可预知的。程序的执行是以“走走停停”的方式运行。

不确定性增加了操作系统的设计与实现难度,操作系统设计者必须采取一定的措施保证系统不出现结果随机性。

6、答:

按层次结构的原则从内到外排列为:裸机、CPU 调度,进程同步操作,内存管理,作业管理,设备管理,文件管理、命令管理和用户。

7、答:

(1) 批处理系统的特点:用户脱机使用计算机,作业成批处理,系统内多道程序并发执行,交互能力差。

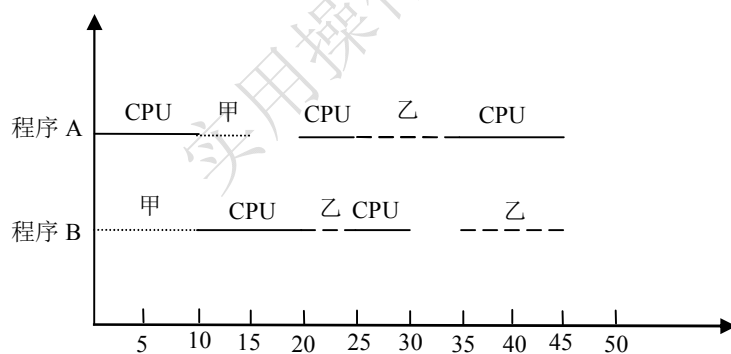
(2) 分时系统的特点:多个用户同时使用计算机,人机交互性强,具有每个用户独立使用计算机的独占性,以及系统响应的及时性。

(3) 实时系统的特点:实时性、可靠性,但系统资源利用率较低。

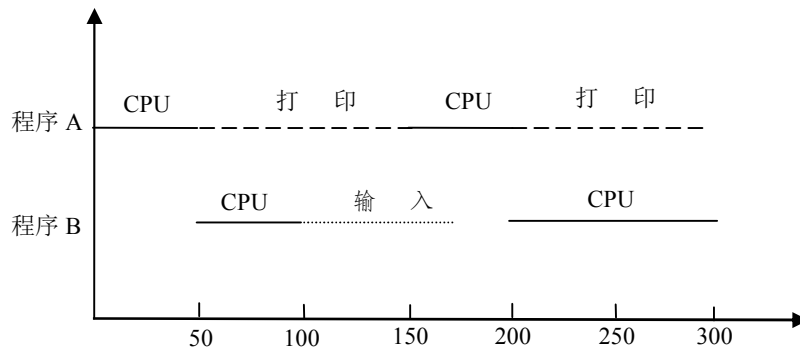
8、答:

顺序执行时,CPU 运行时间为 $(10+5+10)+(10+5)=40\text{s}$,两个程序运行总时间为: $40+40=80\text{s}$,故利用率是 $40/80=50\%$

多道程序环境下,如下图所示,CPU 的运行时间为 40s ,两个程序运行总时间为 45s ,故利用率是 $40/45=88.9\%$



9、答:进程运行情况如下图所示:



(1) CPU 在 100-150ms 时间段内空闲，利用率为 $250/300=83.3\%$

(2) 进程 A 无等待现象。

(3) 进程 B 有等待现象，0-50ms 和 180-200ms。

10、答：便于设计安全可靠的操作系统。核心态和用户态是计算机硬件为保护操作系统免受用户程序的干扰和破坏而设置的两种状态。通常操作系统在核心态下运行，可以执行所有机器指令；而用户程序在用户态下运行，只能执行非特权指令。如果用户程序企图在核心态下运行，只能执行非特权指令。如果用户程序企图在核心态下执行特权指令，将会引起保护性中断，由操作系统终止该程序的执行，从而保护了操作系统。如果允许用户执行特权指令，就有可能干扰操作系统的正常运行，甚至有可能使整个系统崩溃。

11、答：库函数是语言或应用程序的一部分，可以运行在用户空间中。而系统调用是操作系统的一部分，是内核提供给用户的程序接口，运行在内核空间中，而且许多库函数都会使用系统调用来实现其功能。没有使用系统调用的库函数，执行效率通常比系统调用高，因为使用系统调用时，需要上下文的切换以及状态的转换（从用户态转为核心态）。

12、答：从操作系统结构设计观点看，早期的操作系统主要是单处理机操作系统，在结构设计方法上主要采用整体结构设计模式和层次式结构设计模式。

传统的整体式、层次式结构设计法对计算机系统（如网络系统、分布式系统及多处理机系统）而言则有较大缺陷，不能满足需求。后来出现了虚拟机结构操作系统、微内核结构操作系统，对称多处理系统操作系统等。

13、答：

微内核结构操作系统的内核尽量简单，仅存放最基本、最主要的核心功能模块；其他服务和应用建立在内核之上，作为系统进程或用户进程运行。微内核结构操作系统有以下三个主要优点。

- ① 良好扩充性。只需添加支持新功能的服务进程即可增加新功能。
- ② 可靠性高。调用关系明确，执行转移不易混乱。
- ③ 便于网络服务和分布式处理。

习题 2 进程、线程管理

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	B	B	D	C	D	C	A	B	D	C
题号	11	12								
答案	A	A								

二、综合题

2、答：在多道程序环境下，允许多个程序并发执行，这就导致了在操作系统中引入了“进程”。进程是随着操作系统中分时思想的提出而引出的。进程是一个可并发执行的具有独立功能的程序在某个数据集合的一次执行过程，它是操作系统进行资源分配和保护的基本单位。

① 进程和程序的最大区别就是进程是程序的一次执行过程，它是一个动态概念。程序是以文件形式存放在磁盘上的代码序列，它是一个静态概念。

② 进程能够并发执行。在同一段时间内，并发执行的若干进程共享一个处理机，各个进程按照不同的推进速度运行。进程状态及其转换可以很好地描述并发执行进程的执行过程。

③ 进程是计算机系统资源分配的基本单位，程序不能作为一个独立单位运行和申请系统资源。

④ 进程由含有代码和数据的用户地址空间、进程控制块和执行栈区等部分组成，而程序只由静态代码组成。

⑤ 进程和程序之间是多对多的关系。一个程序可被多个进程共用，一个进程在其活动中又可调用若干个程序。

2、答：

原语是由若干条机器指令组成的、用于完成一定功能的一个过程。原语不可分割，其执行期间不允许被中断，要么从头到尾执行一遍，要么全不执行。原语的特征保证其在执行过程中不受外界因素的影响。

原语的一般实现方法是以系统调用的方式提供原语接口，原语在执行过程中采用屏蔽中断的方式来保证其不能被中断。原语常驻内存，只在核心态下运行。通常情况下，原语只提供给系统进程或系统服务器使用。

3、答：

传统操作系统通过进程的并发执行提高了系统资源利用率和作业吞吐量，但进程模型存在如下局限性。

① 每个进程都有一个进程控制块和一个私有的用户地址空间，如果按进程进行并发控制，那么在同一个地址空间中只允许单个执行序列运行。显然，在不进行地址变换的情况下，只允许一个执行序列运行，处理机资源仍然不能得到充分利用。

② 一个进程内部只有一个执行序列，不能满足用户让一个进程内部并发执行多个任务的要求。

③ 进程在处理机上的频繁切换给系统造成大量时空开销，这限制了系统中并发执行进程的数目，降低了系统并发执行程序。

④ 进程通信代价大。进程间传递信息时，要把消息从一个进程的工作区传送到另一个进程的工作区，这需要操作系统提供进程通信机制并且给编程者带来负担。

⑤ 不适合并行计算和分布并行计算的要求。对于多处理机和分布式的计算环境来说，进程之间大量频繁的通信和切换，会大大降低并行度。

线程是进程内部一个相对独立的、具有可调度特性的执行单元。一个进程可包含多个线程。

4、答：内核支持线程（Kernel Supported Threads, KST）是由内核负责管理线程的创建、撤消和切换等，在内核空间为每一个内核支持线程设置一个线程控制块。

内核支持线程实现方式主要有以下优点：

① 多处理器系统中，可以调度使一进程中的多个线程同时执行。

② 提高了线程的并发执行程度，如果进程中的一个线程被阻塞了，内核可调度该进程中的其它线程或其它进程中的线程运行。

③ 内核支持线程具有很小的数据结构和堆栈，线程切换开销小，切换速度快。

④ 内核本身也可采用多线程技术，提高系统并发执行程度。

内核支持线程实现方式的主要缺点是系统需频繁进行用户态和核心态的转换，模式切换开销较大。这是因为用户线程在用户态下运行，但线程的调度和管理由系统内核实现，系统内核负担较大。

用户级线程（User Level Threads, ULT）仅存在于用户空间中，与内核无关。这种线程的创建、撤消、线程间切换、同步和通信等功能，都无需利用系统调用来实现，不需要内核支持。就内核而言，它只是管理常规进程，而感知不到用户级线程的存在。

用户级线程实现的主要优点：

① 不需要得到内核的支持，因此线程开销小，速度快。

② 用户线程和系统线程的调度算法可分开设计，线程库对用户线程的调度算法与操作系统的调度算法无关，线程库可提供多种调度算法供应用程序选择使用。

③ 平台无关性好，用户级线程的实现与系统平台无关。

用户级线程实现的主要缺点：

① 在基于进程机制的操作系统中，内核以进程为单位进行调度，这样如果进程中某一个线程阻塞可能导致整个进程阻塞。

② 在单纯的用户级线程实现方式中，内核每次分派给一个进程仅有一个 CPU，因此无法让同一进程的多个线程在多个处理机上同时运行。

5、答：（1）调度

传统操作系统中，拥有资源的基本单位和独立调度的基本单位都是进程。引入线程的操作系统中，线程作为 CPU 调度的基本单位，真正在处理机上运行的是线程，进程仍作为拥有资源的基本单位。同一进程中的线程切换不会引起进程切换；但一个进程中的线程切换到另外一个进程中的线程时，仍将会引起进程切换。

（2）并发性

引入线程的操作系统中，一个进程可有多个线程，并且线程只能在该进程的地址空间内活动。进程之间的并发执行转变为更多个线程的并发执行，操作系统具有更好的并发性。

（3）拥有资源

不论是传统的操作系统，还是设有线程的操作系统，进程都是拥有资源的一个独立单位。一般地说，线程自己不拥有系统资源（只有一些必不可少的资源），它们共享其所在进程的所有资源。

（4）系统开销

在创建或撤消进程时，系统都要为之分配或回收资源，如内存空间、I/O 设备等。因此，操作系统为此付出的开销将显著地大于创建或撤消线程时的开销。依次类似，在进行进程切换时，涉及到当前进程整个进程运行环境的保存以及新被调度进程的运行环境的恢复。而线程切换时只需保存和设置少量寄存器的内容，并不涉及存储器管理等方面的操作。可见，进程切换开销远大于线程切换开销。此外，由于同一进程中的多个线程具有相同的地址空间，它们之间的同步和通信也比较容易实现。

（5）通信

由于同一进程的线程共享该进程的所有资源，所以不须任何特殊措施就能实现数据共享。而进程通信则相当复杂，必须借助诸如通信机制、消息缓冲、管道机制等措施。

习题 3 进程同步与通信

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	A	D	D	C	B	C	A	B	A	A
题号	11	12								
答案	D	C								

二、综合题

1、答：临界资源也称独占资源、互斥资源，它是指某段时间内只允许一个进程使用的资源。比如打印机等硬件资源，以及只能互斥使用的变量、表格、队列等软件资源。各个进程中访问临界资源的、必须互斥执行的程序代码段称为临界区，各进程中访问同一临界资源的程序代码段必须互斥执行。

为防止两个进程同时进入临界区，可采用软件解决方法或同步机构来协调它们。但是，不论是软件算法还是同步机构都应遵循下述准则：

① 空闲让进。② 忙则等待。③ 有限等待。④ 让权等待。

2、答：忙等待意味着一个进程正在等待满足一个没有闲置处理器的严格循环的条件。因为只有 CPU 为多个进程服务，因此这种等待浪费了 CPU 的时钟。

其他类型的等待：与忙等待需要占用处理器不同，另外一种等待则允许放弃处理器。如进程阻塞自己并且等待在合适的时间被唤醒。忙等可以采用更为有效的办法来避免。例如：执行请求（类似于中断）机制以及 PV 信号量机制，均可避免“忙等待”现象的发生。

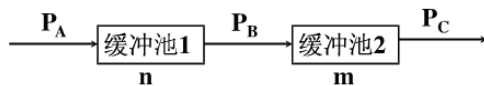
3、答：

在生产者—消费者问题中，Producer 进程中 P(empty) 和 P(mutex) 互换先后次序。先执行 P(mutex)，假设成功，生产者进程获得对缓冲区的访问权，但如果此时缓冲池已满，没有空缓冲区可供其使用，后续的 P(empty) 原语没有通过，Producer 阻塞在信号量 empty 上，而此时 mutex 已被改为 0，没有恢复成初值 1。切换到消费者进程后，Consumer 进程执行

P(full)成功,但其执行P(mutex)时由于Producer正在访问缓冲区,所以不成功,阻塞在信号量mutex上。生产者进程和消费者进程两者均无法继续执行,相互等待对方释放资源,会产生死锁。

在生产者和消费者进程中,V操作的次序无关紧要,不会出现死锁现象。

4、答:



解答: 设置四个信号量empty1、empty2、full1和full2,其同步关系描述如下:

```

int empty1=n; /*表示缓冲池1中的空缓冲区数*/
int empty2=m; /*表示缓冲池2中的空缓冲区数*/
int full1=0; /*表示缓冲池1中装满产品的缓冲区数*/
int full2=0; /*表示缓冲池2中装满产品的缓冲区数*/

```

```

main( )      PA( )
{ cobegin
  PA( );
  PB( );
  PC( );
Coend
}

PB( )
{
  while(1)
  { P(full1);
    从缓冲池1中取出一件产品;
    V(empty1);

    P(empty2);
    将一件产品放入缓冲池2;
    V(full2);
  }
}

PC( )
{
  while(1)
  { P(full2);
    从缓冲池2中取出一件产品;
    V(empty2);
  }
}

```

5、答:

设信号量sp用于控制对盘子的互斥操作,信号量sg1用于计数,表示盘子中的苹果数目,信号量sg2用于计数,表示盘子中的桔子数目。

Semaphore sp=1, sg1=0, sg2=0

```

dad()
{
  while(1)
  { prepare an apple;
    p(sp);
    put an apple on the plate;
  }
}

```

```

v(sg2);}
}
mom()
{
while(1)
{prepare an orange;
p(sp);
put an orange on the plate;
v(sg1);}
}
son()
{
while(1)
{
p(sg1);
take an orange from the plate;
v(sp);
eat the orange;
}
}
daughter()
{
while(1)
{
p(sg2);
take an apple from the plate;
v(sp);
eat the apple;
}
}

```

6、答:

本题是生产者-消费者问题的一个扩展，P1 是生产者，生产的产品分为两种，P2、P3 是消费者，分别消费这两种产品。

(1) 缓冲区是一互斥资源，每次只允许一个进程进入，所以设互斥信号量 mutex，其初始值为 1。

(2) P1、P2 因为奇数的放置与取用而同步，设该信号量为 odd，初始值为 0；P1、P3 因为偶数的放置与取用而同步，设该信号量为 even，初始值为 0，P1、P2、P3 因为共享缓冲区，设同步信号量为 empty，初始值为 N。

semaphore mutex=1,odd=0,even=0,empty=N

```

main()
cobegin{
    Process P1
    while(true)
    { number=produce();

```



```

P(empty);
P(mutex);
put();
V(mutex);
if number%2==0
    V (even) ;
else
    V (odd) ;
}
Process P2
while(true)
{ P(odd);
  P(mutex);
  getodd();
  V(mutex);
  V (empty) ;
  countodd();
}
Process P3
while(true)
{ P(even);
  P(mutex);
  geteven();
  V(mutex);
  V (empty) ;
  counteven();
}
} coend

```

7、答：

Semaphore max; //初始 n+1，表示理发店可以容纳的总人数

Semaphore chair; //初始 n，空闲的椅子

Semaphore barber; //初始 1，表示理发椅空闲

Semaphore finished; //初始 0，表示一次理发结束

Semaphore ready; //初始 0，表示客人准备就绪

Customer:

```

While(1)
{
  wait(max);
  wait(chair);
  wait(barber);
  signal(chair);
  signal(ready);
  ... barbered...
}

```

```

wait(finished);
signal(max);
}
Barber:
While(1)
{
    wait(ready);
    ... barbering...
    signal(finished);
    signal(barber);
}

```

8、答:

本题是生产者-消费者问题的变形。两个生产车间为生产者进程，分别生产 A、B 两种配将；装配车间为消费者，把 A、B 两种产品组装成产品。F1 和 F2 为容量是 10 的缓冲池，需互斥访问。

逐一分析本题中需要同步和互斥的地方，定义信号量及其初值：

- (1) mutex1 用于货架 F1 的互斥访问，初值为 1；
- (2) mutex2 用于货架 F2 的互斥访问，初值为 1；
- (3) empty1 和 full1 用于生产 A 零件车间和装配车间之间的同步制约关系。F1 最多可以放 10 个 A 零件，所以 empty1 为 10；开始时 F1 中没有 A 零件，full1 的初值为 0。
- (4) empty2 和 full2 用于生产 B 零件车间和装配车间之间的同步制约关系。F2 最多可以放 10 个 A 零件，所以 empty2 为 10；开始时 F2 中没有 B 零件，full2 的初值为 0。

A 车间的工作过程的伪代码描述为：

```

while(1){
    生产一个产品 A;
    P(empty1);
    P(mutex1);
    将产品 A 存放到货架 F1 上;
    V(mutex1);
    V(full1);
}

```

B 车间的工作过程的伪代码描述为：

```

while(1){
    生产一个产品 B;
    P(empty2);
    P(mutex2);
    将产品 B 存放到货架 F2 上;
    V(mutex2);
    V(full2);
}

```

装配车间的工作过程伪代码描述为：

```

while(1){
    P(full1);

```

```

        P (mutex1);
        从货架 F1 上取一个 A 产品;
        V (mutex1);
        V (empty1);
        P (full2);
        P (mutex2);
        从货架 F2 上取一个 B 产品;
        V (mutex2);
        V (empty2);
        将取得的 A 产品和 B 产品自组装成产品;
    }

```

9、答：

本题中共有三类进程，他相当于机房管理员进程 guard，学生进程 student 和教师进程 teacher。

相应的信号量和各个进程描述如下：

```

semaphore computer=2m;    /*对应于计算机的资源信号量*/
semaphore student=0;      /*对应于欲进入机房的学生*/
semaphore enter=0;        /*用来控制学生是否可进入机房*/
semaphore finish=test=0;  /*用来同步学生和教师——教师须检查实习完毕的学生*/

```

```

student_i() {                /*i=1, 2, ... 2n*/
    V(student);              /*激活管理员，有学生到达，要进入机房实验*/
    P(enter);                /*等待管理员激活进入机房*/
    进入机房上机实习;
    V (finish);              /*激活教师已经做完实验*/
    P(test);                 /*等待教师检查作业*/
    离开机房;
    V (computer);            /*所占用的计算机变为空闲*/
}

guard() {
    int i;
    for(i=0;i++;i<n) {
        P(computer);        /*等待有两个空闲计算机*/
        P(computer);
        P(student);          /*等待有两个学生达到*/
        P(student);
        V(enter);            /*激活两个等待进入机房的学生*/
        V(enter);
    }
}

teacher() {
    int i;
    for(i=0;i++;i<n) {
        P (finish) ;        /*等待两个学生完成实验*/

```

```

        P(finishi);
        检查两个学生的实习结果;
        V(test);          /*检查完后，激活两个学生检查完毕，可以离开机房*/
        V(test);
    }
}

```

10 答：本题有两个临界资源：一个是出入口，另一个是博物馆。

本题需要定义两个信号量

```
semaphore empty=500;
```

```
semaphore mutex=1;
```

```
cobegin
```

```
参观者进程 i;
```

```
{ ...
```

```
    P(empty);
```

```
    P(mutex);
```

```
    进门;
```

```
    V(mutex);
```

```
    参观;
```

```
    P(mutex);
```

```
    出门;
```

```
    V(mutex);
```

```
    V(empty);
```

```
    ...
```

```
}
```

```
Coend;
```

11、答：

虽然信号量机制是一种有效的进程同步机制，但其存在以下缺点：

① 信号量的 P、V 操作由用户在各个进程中分散使用，使用不当容易造成死锁，增加了用户编程负担。

② 信号量机制涉及多个程序的关联内容，程序代码可读性差。

③ 使用信号量机制不利于代码的修改和维护，程序模块独立性差，任一变量或一段代码的修改都可能影响全局。

④ 信号量机制的正确性很难保证。

管程由四部分组成：

① 管程的名称；

② 局部于管程内部的共享数据结构说明；

③ 对该数据结构进行操作的一组过程；

④ 对管程内部共享数据设置初始值的语句。

12、答：

管程与进程不同，主要体现在以下方面：

- ① 定义的数据结构及其在数据结构上进行的操作不同。进程定义的是私有数据结构，主要对数据进行处理运算；管程定义的是公共数据结构，主要进行同步和初始化操作。
- ② 目的不同。进程在于实现系统的并发性，管程是为了解决共享资源的互斥使用。
- ③ 进程通过调用管程中的过程实现对共享数据结构的操作，进程为主动工作方式，管程为被动工作方式。
- ④ 进程之间能并发执行，而管程则不能与调用进程并发执行。
- ⑤ 进程具有动态性，而管程则是操作系统中一个资源管理模块，供进程调用。

13、答：高级通信机制可分为三大类：

1. 共享存储器系统

共享存储器系统中相互通信的进程间共享某些数据结构或存储区，彼此能够通过这些空间进行通信。

2. 消息传递系统

进程间的数据交换常以格式化的消息为单位。在计算机网络中，又把消息称为报文。程序员直接利用系统提供的一组通信命令进行通信。消息传递系统按实现方式分成两种。

3. 管道通信

所谓“管道”是指用于连接一个读进程和一个写进程以实现它们之间通信的一个共享文件。向管道(共享文件)提供输入的发送进程(即写进程)以字符流形式将大量的数据送入管道；接受管道输出的接收进程(即读进程)从管道中接收(读)数据。

14、答：P、V 操作是指进程之间通过共享变量实现信息传递，而高级通信机制是由系统提供发送（send）与接收（receive）两个操作，进程间通过这两个操作进行通信，无须共享任何变量。

15、答：基本原理：操作系统管理一个用于进程通信的缓冲池，其中的每个缓冲区单元可存放一条消息。发送消息时，发送者从中申请一个可用缓冲区，接收者取出一条消息时再释放该缓冲区，每个进程均设置一条消息队列，任何发送给该进程的消息均暂存在其消息队列中。

习题 4 处理机调度

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	D	D	C	A	D	D	C	C	B	C
题号	11	12	13	14	15					
答案	A	A	D	C	C					

二、综合题

1、答：

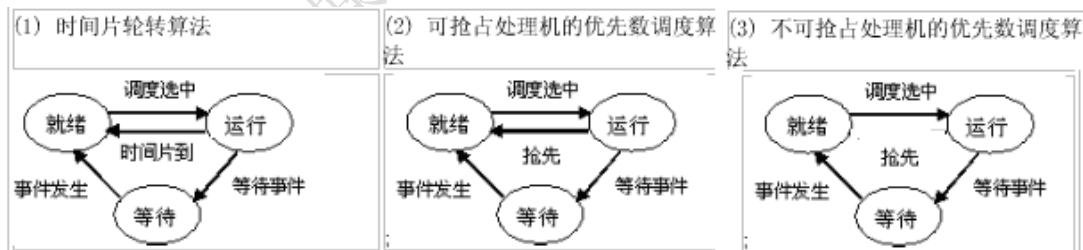
(1) 处理机的三级调度是指一个作业在运行过程中要遇到的高级调度（作业调度）、中级调度（进程对换）、低级调度（进程调度）。不过不是所有的操作系统都有这三级调度，有些只实现了其中的一个或两个，但是每个操作系统都有进程调度。

(2) 高级调度主要在需要从外存调入一个作业到内存中时发生；中级调度主要在内存紧张需要调出一些进程或者内存空闲需要把先前调出的进程调回内存时发生；低级调度主要在正在执行的进程放弃 CPU 或者其他优先级高的进程抢占 CPU 的时候发生。

(3) 高级调度主要工作是决定外存的后备队列中哪个进程被调入内存中，并给这个作业创建进程，分配它必要的资源；中级调度主要工作是在内存紧张的时候把就绪队列中暂时得不到执行的进程换出到外存，也负责在内存比较空闲时把换到外存的进程调回到内存；低级调度主要工作是决定把 CPU 分配给就绪队列中的哪个进程。

2、答：作业调度和进程调度。作业调度时宏观调度，它决定后备队列中哪些作业能进入内存运行。进程调度是微观调度，它决定进入内存的这些作业中的哪一个进程占有处理器。作业调度是高级调度，它位于操作系统的作业管理层次。进程调度是低级调度，它位于操作系统分层结构的最内层。

3、答：



4、答：

(1) 只要有就绪进程，进程调度程序便必将从其中选择优先权最大的进程投入执行。故没有运行进程，则肯定就没有就绪进程。

(2) 当系统中只有一个进程且它具备执行条件时；或者，系统中只有一个进程具备执行条件，而其他的进程均处于阻塞状态时，系统中便会出现只存在运行进程却没有就绪进程的现象。如果系统中的所有进程均处于阻塞状态（例如它们都在等待 I/O 操作的完成），则系统中便会出现既没有运行进程也没有就绪进程的现象。

(3) 如果系统采用立即抢占的优先权调度方式，则运行进程将是自由进程中优先权最高的（不考虑切换过程中的暂时状态）；否则，运行进程的优先权可能比某个就绪进程低。

5、答：

对终端型作业用户而言，他们提交的作业大多属于交互型作业，作业通常较小，系统只要能使这些作业在第一个队列所规定的时间内完成，便可使他们都感到满意。对短作业用户而言，开始时他们的作业像终端型一样，如果仅在第一个队列中执行一个时间片即可完成，便可获得与终端型用户作业一样的响应时间；对于稍长的作业，通常也只需在第二个队列和第三队列各执行一个时间片即可完成，其周转时间仍然很短。对长批处理作业用户而言，他们的作业将依次在第 1, 2, ..., n 个队列中运行，然后再按轮转方式运行，用户不必担心其作业长期得不到处理；而且每往下降一个队列，其得到的时间片将随着增加，故可进一步缩短长作业的等待时间。

6、答：

(1) 采用先来先服务调度算法，则其调度顺序是 1、2、3、4。

作业号	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10.0	2.0	10.0	12.0	2.0	1.0
2	10.2	1.0	12.0	13.0	2.8	2.8
3	10.4	0.5	13.0	13.5	3.1	6.2
4	10.5	0.3	13.5	13.8	3.3	11.0

平均周转时间： $T=(2.0+2.8+3.1+3.3)/4=2.8$

平均带权周转时间： $W=(1.0+2.8+6.2+11.0)/4=5.25$

(2) 采用短作业优先调度算法，则其调度顺序是 1、4、3、2。

作业号	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10.0	2.0	10.0	12.0	2.0	1.0
2	10.5	0.3	12.0	12.3	1.8	6.0
3	10.4	0.5	12.3	12.8	2.4	4.8
4	10.2	1.0	12.8	13.8	3.6	3.6

平均周转时间： $T=(2.0+1.8+2.4+3.6)/4=2.45$

平均带权周转时间： $W=(1.0+6.0+4.8+3.6)/4=3.85$

7、答：

各个作业执行的时间如下图所示：

	8:00	8:10	8:20	8:30	8:40	8:50	9:00	9:10	9:20	9:30
1	深黑	深黑	深黑	深黑	深黑	深黑	深黑	深黑	深黑	深黑
2										
3										
4										
5										

注：深黑色表示作业独占 CPU 时间，浅黑色表示作业平分 CPU 时间，白色表示 CPU 空闲时间。

(1) 作业调度顺序是：1, 3, 4, 2, 5。

(2) 最大作业周转时间为 55 分钟。

(3) 全部作业运行结束的时刻为：9: 30

8、答：

8: 00 时,作业 1 运行;

9: 10 时, 作业 1 运行完毕, 其他 3 个作业均已到达, 它们的响应比分别为:

$$R_2=1+(9:10-8:40)/30=2$$

$$R_3=1+(9:10-8:50)/10=3$$

$$R_4=1+(9:10-9:10)/5=1$$

作业 3 先运行，10 分钟后，作业 3 运行完毕。

9:20 时，作业 3 运行完毕，其他两个作业的响应比是：

$$R_2 = 1 + (9:20 - 8:40) / 30 = 2.3$$

$$R_4 = 1 + (9:20 - 9:10) / 5 = 3$$

作业 4 先运行，5 分钟后，作业 4 运行完毕，最后作业 2 运行。

作业的执行顺序为：1、3、4、2。

实用操作系统教程（第2版）

习题 5 死锁

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	C	B	B	B	D	B	D	A	C	C
题号	11	12	13	14						
答案	B	B	D	C						

二、综合题

3、答：

死锁产生的 4 个必要条件是：

(1) 互斥条件 (2) 请求和保持条件 (3) 不剥夺条件 (4) 环路等待条件

解决死锁的常用措施有：

- (1) 预防死锁
- (2) 避免死锁
- (3) 检测死锁
- (4) 解除死锁

2、答：

$$(1) \text{Need} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 6 & 4 & 2 \end{bmatrix}$$

(2) 因为存在一个安全序列 $\langle P_0, P_2, P_1, P_3 \rangle$ ，所以系统处于安全状态。

(3) 先试着满足 P_1 进程的要求，存在一个安全序列 $\langle P_0, P_2, P_1, P_3 \rangle$ ，所以系统仍处于安全状态，所以 P_1 的这个请求应该马上满足。

3、答：当 $N \leq 3$ 时没有死锁的危险。当 $N=4$ 时，可能出现四个进程各自占有 2 台磁带机，又各自申请 1 台磁带机的情况，这样就出现了死锁，若 $N>4$ ，死锁的可能性就更高了。

4、答：

若不加限制，可能会发生死锁，例如：进程 P_1 、 P_2 和 P_3 分别获得资源 R_3 、 R_1 和 R_2 后再继续申请资源时都要等待，这是循环等待。解决方法可有几种：

(1) 采用静态分配

由于执行前已获得所需的全部资源，故不会出现占有资源又等待别的资源的现象。

(2) 采用按序分配

不会出现循环等待资源现象。

(3) 采用银行家算法

因为在分配时，保证了系统处于安全状态。

5、答：

(1) 可能发生死锁。当两个进程各自均已得到 1 个资源时，这两个进程都将会因申请第 2 个资源的请求无法得到满足而阻塞，这时，它们便进入死锁状态。

(2) 不会发生死锁。因为系统中只有 2 个进程，每个进程最多需要 2 个资源，而系统中总共有 3 个资源，因此必定有一个进程可成功申请到 2 个资源，待它运行完毕归还资源后，另一个进程也可顺利运行并完成。

(3) 可能发生死锁。当其中一个进程得到 1 个资源，而另一个进程得到 2 个资源时，这两个进程均需进一步申请资源才能继续运行，并会因为得不到满足而阻塞，这时，它们将进入死锁状态。

(4) 不会发生死锁，因为总共有 3 个进程，每个进程最大需求为 2，而资源总量为 5，故必有一个进程可顺利申请到 2 个资源，在它完成并释放资源后，其他进程也可顺利完成。

(5) 可能发生死锁。当每个进程各自获得 2 个资源后，该类资源便已被全部分配出去，而每个进程的运行都需要进一步申请资源，并将因系统无空闲资源而阻塞，从而使系统进入死锁状态。

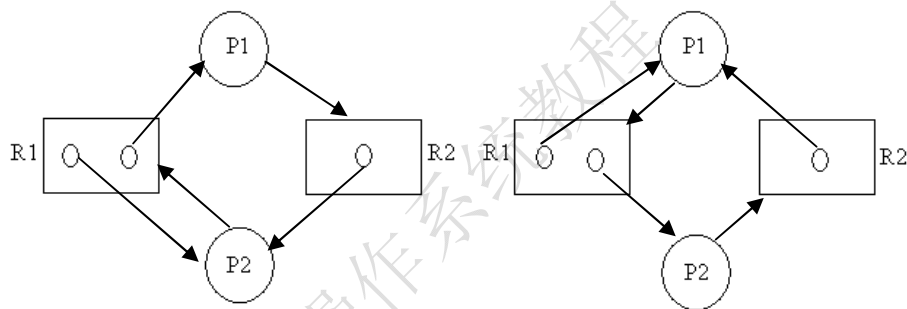
6、答所谓按序分配是指，系统将所有的资源按类型进行排序，并给不同的类型赋予不同的序号，而所有的进程对资源的请求，必须严格按资源序号递增（或递减）的顺序提出。

在采用按序分配方式时，如果系统要求进程严格按照资源序号递增的顺序来申请资源，则只可能存在拥有较低序号资源的进程等待拥有较高序号资源的进程释放资源的现象，而不会存在相反的等待，因此，死锁产生的必要条件之一——“环路等待”条件不可能成立，从而达到防止死锁发生的目的。

7、答：

当两个进程各申请到一个 R1 资源后，系统便进入不安全状态。此时，无论谁先获得 R2 资源，它都将因申请另一个 R1 资源而阻塞，并等待另一个进程释放 R1 资源。而另一个进程则将等待前一进程释放 R2 资源。因此进入死锁状态。

死锁点资源分配图如下图所示。



C、当一个进程申请资源时，如果系统已无此类资源可用，但某个拥有该资源的其他进程处于阻塞状态，则允许前者抢占后者的资源。这样可破坏“不剥夺”条件。

习题 6 内存管理

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	B	A	D	B	A	B	A	B	C	B
题号	11	12	13	14	15					
答案	B	A	A	C	C					

二、综合题

4、答：操作系统中的存储管理主要指内存管理。内存又称主存，它是计算机系统中仅次于 CPU 的另一个宝贵资源。内存的主要职责是存放程序、数据以及操作结果，任何程序只有装入内存后才能被处理机执行，管理好内存是操作系统的重要任务之一。

2、答：（1）内存分配和回收：记录内存的使用情况，为每道程序分配内存空间，回收系统或用户程序释放的内存空间。

（2）内存保护：确保每道用户程序都只在自己的内存空间内运行，彼此互不干扰。

（3）地址变换：将地址空间中的逻辑地址转换为内存空间中与之对应的物理地址。

（4）内存扩充：借助于虚拟存储技术来扩大物理内存的容量，使用户所感觉到的内存容量比实际内存容量大得多。

3、答：

当一个程序的相对地址装入到与其逻辑地址空间不一致的绝对地址空间中时，为了保证程序的正确运行，必须把指令和数据的逻辑地址转换为物理地址，这项工作称为地址重定位。

① 静态地址重定位

在程序装入时由程序装入程序（装配程序）实现地址转换，将程序中的所有逻辑地址都加上目标代码在主存内的起始地址。这种方式要求地址变换在程序执行前一次性完成。

② 动态地址重定位。

程序执行过程中，CPU 在执行指令时实现地址转换。在多道程序系统中，内存空间常常被多个进程共享，程序员事先不可能知道程序执行时在内存中的物理位置，且必须允许进程在执行期间因对换或空闲区拼接而移动，这都需程序的动态重定位。动态重定位通常利用基址寄存器的内容加上变址寄存器中的内容计算出指令的物理地址，它需要借助一定的硬件地址转换机构才能实现。

4、答：

当某一个进程执行完成并释放所占分区时，系统应进行回收，此时会出现以下 4 种情况。

① 若回收区只与上空闲区相邻接，即其低地址部分邻接一空闲区。此时将回收区与上空闲区合并，不必为回收区分配新表项，只需修改上空闲区的大小为二者之和即可。

② 若回收区只与下空闲区相邻接，即其高地址部分邻接一空闲区。此时将回收区与下空闲区合并，不必为回收区分配新表项，只需修改下空闲区的起始地址为回收区的起始地址、大小为二者之和。

③ 若回收区与任何空闲区均不相邻接，即其高、低地址部分都不邻接一个空闲区。此时则需要为回收区建立一个新的表项，填写回收区的大小和起始地址，并将其插入空闲分区表（链）中相应的位置。

④ 若回收区与上、下空闲区均邻接，即其高、低地址部分均邻接一个空闲区。此时三个邻接的空闲区合并，即将下空闲区表项在空闲区表中删除，修改上空闲区表项中的长度为三者之和即可。

5、答：（1）采用首次适应算法，在完成了题目所给的系列申请及释放内存操作后，空闲分区如下所示。

分区	大小	起始地址
0	30K	150K
1	20K	280K
2	112K	400K

（2）采用最佳适应算法，在完成了题目所给的系列申请及释放内存操作后，空闲分区如下所示。

分区	大小	起始地址
0	30K	400K
1	42K	470K
2	90K	210K

（3）采用最差适应算法，在完成了题目所给的系列申请及释放内存操作后，空闲分区如下所示。

分区	大小	起始地址
0	30K	150K
1	80K	220K
2	52K	460K

（4）如再申请 100K，有上述结果可知，采用首次适应算法后剩下的空闲分区能满足这一申请要求，采用最佳适应算法和最差适应算法均不能满足申请要求。

6、答：（1）程序空间的大小为 32KB，因此逻辑地址的有效位数是 15 位。

（2）内存储空间的大小是 16KB，因此物理地址至少需要 14 位。

（3）当页面为 1KB 时，虚地址 0A5C 表示页号为 00010，页内地址是 1001011100。该页在内存的第 4 块，即块号为 0100，因此 0A5C 的物理地址是 01001001011100，即 125CH。用同样的方法可以求得，053C 的物理地址是 293CH，103C 的逻辑地位在第 4 页，产生越界异常。

7、答：（1） $1.5 \times 2 = 3$ 微秒 （2） $1.5 \times 2 \times 15\% + 1.5 \times 85\% = 1.725$ 微秒

8、答：（1）5 位页号+11 位页内偏移

（2）进程的页表最多是 32 项，每项为 9 位（ $1M/2K=2^9$ ）。

（3）进程页表项不变，还为 32 项目，每项为 8 位（ $0.5M/2K=2^8$ ）。

9、答：具有两级页表的页式存储管理的地址空间依然是一维的，两级页的划分对于进程来说都是透明的。而段页式存储管理的地址空间是二维的，段的划分用户能感觉到。

10、答：逻辑地址（0，137），（1，4000），（2，3600），（5，230）中的 0，1，2，5 表示段号，137，4000，3600，230 表示位移量。段号 0 中的内存地址 50K 表示 0 号段的起始地址，10K 表示这个段的长度。对于逻辑地址（0，137），先找到段号 0 处，物理地址=起始地址+位移量，即物理地址=50K+137=50*1024+137=51337。（说明 1K=1024）

对于（1，4000），由于位移量 4000>3*1024，所以越界，产生中断信号

对于（2，3600），找到段号 2 处，物理地址=70K+3600=70*1024+3600=75280

对于（5，230），逻辑地址的段号 5>段表长度 4，发生越界，产生中断信号。

11、答：在段式存储管理中，段的长度不能大于内存的长度，因为一个独立的段占用一段连续的内存空间，内存分配是以段为单位进行的，如果一个段的长度大于内存的长度，那么该段将无法调入内存。在段页式存储管理中，段的长度可以大于内存的长度。因为内存分配的

单位是页，一个段内逻辑上连续的页面，可以分配到不连续的内存页面中，不要求一个段的所有逻辑页都进入内存。

12、答：(1)在页式存储管理中，分页对于用户是透明的，一个进程由若干个页构成，所有页的长度相同；(2)在段式存储管理中，分段对于用户是可见的，一个进程由若干个段构成，各个段的长度可以不同，一个段恰好对应一个程序单位；(3)在段页式存储管理中，段的划分对用户是可见的，段内页的划分对用户是透明的，一个段由若干个页构成，所有页的长度相同。

13、答：页式存储管理优缺点：(1)静态等长存储分配简单，有效地解决了内存碎片问题；(2)共享和保护不够方便。段式存储管理优缺点：(1)动态异长存储分配复杂，存在碎片问题；(2)共享与保护方便；(3)可以实现动态链接和动态扩增。

实用操作系统教程（第2版）

习题 7 虚拟存储管理

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	A	B	B	C	A	A	B	B	A	A
题号	11	12	13	14	15	16				
答案	C	B	B	B	C	B				

二、综合题

1、答：把一个程序分为一系列功能相对读了的程序单元（称为覆盖），让执行时并不要求同时装入内存的覆盖组成一组（称为覆盖段），共享同一个存储区域，这种内存扩充就是覆盖。交换技术就是把暂时不用的某个程序及数据部分或全部从内存移到外存中去，以便腾出必要的内存空间，或把制定的程序或数据从外存读到相应的内存中，并将控制权转给它，让其在系统上运行的一种内存扩充技术。

覆盖技术要求程序员必须把一个程序划分成不同的程序段，并规定好它们的执行和覆盖顺序，操作系统根据程序员提供的覆盖结构来完成程序段之间的覆盖。覆盖主要在同一个作业或同一个进程内进行；而交换主要是在进程或作业之间进行。另外，覆盖只能覆盖那些与覆盖程序段无关的程序段。

2、答：

不对。交换是把内存中暂时不能运行的进程或暂时不用的程序和数据换出到外存上，以腾出足够的内存空间，把已具备运行条件的进程或进程所需的程序、数据换入内存。交换是提高内存利用率的有效措施。

而虚拟存储器是指仅把作业的一部分装入内存便可运行作业的存储器系统，是指具有请求调入功能和置换功能，能从逻辑上对内存容量进行扩充的一种存储器系统。虚拟存储系统的实现，都是建立在离散分配存储管理的基础之上。虽然交换能提高内存利用率，但仅使用交换技术，仍然不能实现仅把作业的一部分装入内存便运行作业，故交换并不能实现虚拟存储器。

3、答：

为了给大作业（其地址空间超过主存可用空间）用户提供方便，而由操作系统把主存和辅存统一管理起来并实现自动交换。即一个大作业在执行时，一部分地址空间在主存，而另一部分在辅存，当访问的信息不在主存时由操作系统（而不是有程序设置的 I/O 指令）将其从辅存调入主存。从效果上看，该计算机系统好像为用户提供了一个容量比主存大得多的存储器，这个存储器称为虚拟存储器。

虚拟存储器的基本特征是：①虚拟扩充，即不是物理上而是逻辑上扩充了内存容量；

②部分装入，即每个作业不是全部一次性地装入内存，而是只装入一部分；③离散分

配，即不必占用连续的内存空间，而是“见缝插针”；④多次对换，即所需的全部程序和数据要分成多次调入内存。

4、答：

0x2C27: 0x7C27

0x1D71: 缺页

0x4000: 越界

5、答：m 条指令实际花费时间应为执行 m 条指令花费的时间与操作系统处理一次页故障需要时间之和。一条指令执行平均需要 k (ns)，m 条指令执行需要 m * k (ns)，执行 m 条指令发生一次缺页中断，需要 n (ns)，也即 m 条指令实际花费时间为 (m * k + n) (ns)，则

平均每条指令的执行时间为 $(m * k + n) / m$ (ns)。

6、答：

LRU 方法：10 次，FIFO 方法：14 次，Optimal 方法：8 次。

7、答：

(1) 由表中装入时间可知，最先装入内存的是第 3 页，所以采用 FIFO 算法时，将选择第 3 页进行置换。

(2) 由表中最后访问时间可知，最近最久未使用的是第 2 页，所以采用 LRU 算法时，将选择第 2 页进行置换。

(3) 从表中可知，访问位和修改位均为 0，即未访问过的页面，又未被修改过的页面是第 0 页，所以采用改进的 Clock 算法时，将选择第 0 页进行置换。

8、答：

在虚拟存储器系统中，刚被淘汰的页面又马上被调回内存，调回内存不久又马上被淘汰出去，如此反复的局面，这种现象称为抖动（或称颠簸）现象。如果在页面置换算法设置不好的情况下，就会出现同一个页面频繁地在主、辅存之间的“抖动”现象。

直观上，分配给进程的物理块越多，进程执行时发生的缺页次数就越小。在某些情况下，当分配的物理块多反而导致缺页次数更大，这种奇怪的现象称为 Belady 异常现象。例如对于页面引用串 1,2,3,4,1,2,5,1,2,3,4,5。在内存中分配了 3 个物理块时发生了 9 次缺页中断，而当分配了 4 个物理块时发生了 10 次缺页中断。这种异常现象只会在 FIFO 算法里可能出现，不会在其它算法里出现。

9、答：

在指令中如果包含地址部分，则必须进行地址变换，同时进行越界检查和权限检查，只在两者均合法时，才完成指令规定操作。

(1) 由于第 0 段的存在位为 0，表示该段未装入内存，因此产生缺段中断。

(2) 从段表第 1 项可看到，指令中逻辑地址合法，段也已经在内存，但存取控制字段不符，故产生保护性中断信号。

(3) 逻辑地址合法，存取方式合法，形成物理地址 8020 后，执行指定操作。

(4) 逻辑地址中段内地址超长，产生越界中断信号。

(5) 逻辑地址及访问方式合法，形成物理地址 3100，指令执行后，将条转到内存单元 3100 处继续执行。

10、答：

(1) 一个作业最多可以有 $2^8=256$ 个段。

(2) 每段的最大长度为 $2^{16}=64KB=65\ 536$ 字节。

(3) 逻辑地址[0, 430]主存地址为：2100+430=2530；

逻辑地址[1, 50]无法进行地址变换，因为产生了越界中断；

逻辑地址[2, 30]无法进行地址变换，因为产生了缺段中断；

逻辑地址[3, 70]的主存地址为：4000+70=4070。

11、答：(1) 用更快的 CPU，没用。

(2) 用更大的磁盘没用，交换空间本来就足够了。

(3) 增加多道程序的道数，没用，情况会更糟。

(4) 减少并发进程数，效果明显。

(5) 用更快的其他 I/O 设备，效果不明显。

12、答：

(1) 页面大小为 4KB，故页内偏移为 12 位。系统采用 48 位虚拟地址，故虚页号为 $48-12=36$

位。采用多级页表时，最高级页表项不能超出一页大小，故应采用4级页表，最高级页表项正好占据一页空间。

(2) 系统进行页面访问操作时，首先读取页面对应的页表项，有98%的概率可以在TLB中直接取到，然后进行地址转换，如果TLB为命中，则要通过一次内存访问来读取页表项。页面的平均访问时间为： $98\% * (10+100) + (1-98\%) * (10+100+100) = 112\text{ns}$

(3) 二级页表的平均访问时间计算同理：

$$98\% * (10+100) + (1-98\%) * (10+100+100+100) = 114\text{ns}$$

(4) 设快表命中率为P，则应满足：

$$P * (10+100) + (1-P) * (10+100+100+100) \leq 120\text{ns}, \text{解得: } P \geq 95\%$$

(5) 系统采用48位虚地址，每段最大为4G，故段内地址为32位，段号： $48-32=16$ 位。每个用户最多可以有 2^{16} 个段，段内采用页式地址，与(1)中计算同理， $(32-12)/9$ ，取上整为3，故段内应采用3级页表。

13、答：程序1按行优先的顺序访问数组元素，与数组在内存中存放的顺序一致，每个内存页面可存放200个数组元素，这样，程序1每访问两行数组元素产生一次缺页中断，所以程序1的执行过程会发生50次缺页。

程序2按列优先的顺序访问数据元素，由于每个内存页面存放两行数组元素，故程序2每访问两个数组元素就产生一次缺页中断，整个执行过程会发生5000次缺页。

若每页只能存放100个整数，则每页仅能存放一行数组元素，同理可以计算，程序1的执行过程产生100次缺页，程序2的执行过程产生10000次缺页。

以上说明缺页的次数与内存中数据存放的方式及程序执行的顺序有很大关系；同时说明，当缺页中断次数不多时，减小页面大小影响并不大，但缺页中断次数很多时，减小页面大小会带来很严重的影响。

习题 8 I/O 设备管理

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	B	D	B	A	A	D	D	B	B	B
题号	11	12	13	14	15	16	17	18	19	20
答案	A	C	A	A	A	A	A	C	C	A

二、综合题

- 5、答：DMA 是 Direct Memory Access（直接存储器访问）的缩写。DMA 方式的特点是，数据传输的基本单位是数据块，所传输的数据时从设备直接送入内存，期间不需要 CPU 的干预，或者相反；仅在传送一个或多个数据块的开始和结束时才需要 CPU 的干预，整块数据的传送是在 DMA 控制器的控制下完成的。

DMA 与中断方式的主要区别是：中断驱动 I/O 控制方式每几个数据传输后即发出一次中断，DMA 控制方式是在一批数据传输完成后发出一次中断；中断驱动 I/O 控制方式下数据的传输是由 CPU 控制的，DMA 控制方式下在数据块传输的开始和结束阶段由 CPU 控制，在传输过程中由 DMA 控制器控制。

2、答：

（1）和（3）为设备驱动程序实现。（2）和（4）为逻辑 I/O 层实现。

- 3、答：通道是一种特殊的 I/O 处理机，它具有执行 I/O 指令的能力，并通过执行通道(I/O)程序来控制 I/O 操作。但 I/O 通道又与一般的处理机不同，主要表现在以下两个方面：一是其指令类型单一；二是通道没有自己的内存，与 CPU 共享内存。

通道经常采用交叉连接是为了增加通路，即使得每一个设备与所有的控制器相连，每个控制器与所有的通道相连，增加了存储器与设备之间的通路，这样在设备分配时，可选择范围就大。

- 4、答：（1）先来先服务算法的调度顺序为：20，44，40，4，80，12，76

移动的柱面数分别为：20，24，4，36，76，68，64

柱面移动总量为：292

寻道时间为：292*3 ms=876 ms

- （2）最短寻找时间优先算法调度顺序为：40、44、20、12、4、76、80

移动的柱面数分别为：0，4，24，8，8，72，4

柱面移动总量为：120

寻道时间为：120*3ms=360ms

- 5、答：每条记录的读取时间为 20ms/4 = 5ms，优化前处理总时间为：

$$[(5+5) + (5*3+5+5) + (5*3+5+5) + (5*3+5+5)] \text{ms} = 85 \text{ ms}$$

优化后记录顺序为：A，C，B，D。

A	C	B	D
---	---	---	---

优化后处理总时间=(5+5)【处理 A】+(5+5)【处理 B】+5【跳过 A】+(5+5)【处理 C】+ (5+5)【处理 D】= (5ms + 5ms) *4+5=45 ms

- 6、答：（1）电梯调度算法的调度的次序为：65、71、100、114、115、49、40、36

（2）最短寻找时间优先算法的调度的次序为：65，71，49，40，36，100，114，115

（3）循环扫描调度算法的调度的次序为：65、71、100、114、115、36、40、49

- 7、答：（1）采用 FCFS 算法调度时，磁头移动顺序为：

149、86、147、91、177、94、150、102、175、130

磁头移动总距离为：

$$(149-86)+(147-86)+(147-91)+(177-91)+(177-94)+(150-94)+(150-102)+(175-102)+(175-130) \\ =571 \text{ 条磁道}$$

(2) 采用 SSTF 算法调度时，磁头移动顺序为：

149、147、150、130、102、94、91、86、175、177

磁头移动总距离是 168 条磁道。

(3) 采用 SCAN 算法调度时，磁头移动顺序为：

149、147、150、175、177、199、130、102、94、91、86

磁头移动总距离是 261 条磁道。

8、答：(1) $1*0.25+100*(1-0.25)=75.25\text{ms}$

(2) $1*0.5+100*(1-0.5)=50.5\text{ms}$

9、答：(1) 先来先服务算法的调度顺序为：10，22，20，2，40，6，38

移动的柱面数分别为：10，12，2，18，38，34，32

柱面移动总量为：146

寻道时间为： $146*6 \text{ ms}=876 \text{ ms}$

(2) 下一个最邻近柱面算法调度顺序为：20，22，10，6，2，38，40

移动的柱面数分别为：0，2，12，4，4，36，2

柱面移动总量为：60

寻道时间为： $60*6\text{ms}=360\text{ms}$

10、答：(1) 读取并处理完 A 记录后磁头移动到 3、4 交接处，花费时间为 $2+4=6\text{ms}$ ，接下来每次读取一个记录，都是先移动 8 个扇区位置在进行读取和处理，故总共花费时间为 $(2+4)+9*(2+4+2*8)=204\text{ms}$ 。

(2) 优化记录如下：

扇区号	1	2	3	4	5	6	7	8	9	10
逻辑记录	A	H	E	B	I	F	C	J	G	D

如此优化后每次读取并处理完记录后磁头刚好移动至下一个要读取的数据处，优化后供需时间为 $10*(2+4)=60\text{ms}$ 。

11、答：它们都是要尽量减少移动臂时所花的时间。不同点：“最短寻找时间优先算法”不考虑臂的移动方向，总是优先选择离当前位置最近的那个柱面的访问者，这种选择可能导致移动臂来回改变移动方向；“电梯调度算法”是沿着臂的移动方向去选择，仅当沿臂移动方向无等待访问者时才改变臂的移动方向。

12、答：因为一个字符占 10 位，因此在 56Kbps 的速率下，每秒传送： $56000/10=5600$ 个字符，即产生 5600 次中断。每次中断需 0.1ms，故处理调制器占用 CPU 时间总共为 $5600*0.1\text{ms}=560\text{ms}$ ，占用 56%CPU 时间。

习题 9 文件系统

一、选择题

题号	1	2	3	4	5	6	7	8	9	10
答案	D	A	D	B	B	D	B	B	B	C
题号	11	12	13	14	15	16	17	18	19	20
答案	B	B	B	D	D	A	EB	A	C	C
题号	21	22	23	24	25					
答案	A	A	A	B	B					

改错：选择第 9 题题干：从**用户**的观点看...

二、综合题

6、答：文件系统中要设置打开文件和关闭文件操作。原因是：用户进程存取一个文件，系统首先要检索目录结构，按名查找该文件的文件控制块。打开文件的基本思想是，按指定文件名检索目录结构，把找到的文件控制块读入并保存到内存，此后每次存取该文件时，就无需再执行按名查找的过程，可以直接在内存找到它的文件控制块，从而加快了存取速度。文件打开后，可以对该文件进行读、写等操作。当一个文件不再存取时，需要关闭该文件，释放占用的系统资源，并将文件控制块的内容复制到外存储器中。这样做，一方面提高了资源利用率，另一方面保证了数据的安全。

2、答：2301/1024 商为 2，余数为 253。要访问字节的逻辑记录号为 2，对应的物理磁盘块号为 75。故应访问第 75 号磁盘块。

3、答：

假设每次盘 I/O 的大小为一个盘块，即 512 字节。

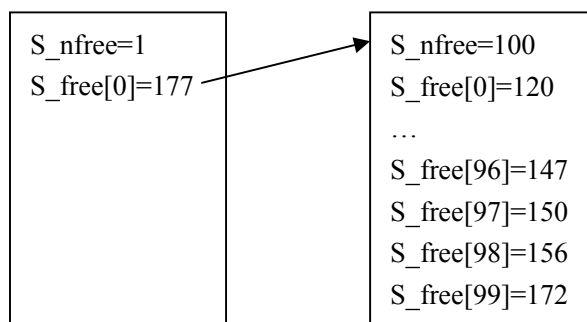
(1) 由于根目录的第一块常驻内存，根目录找到文件 A 需要 5 次读盘。因为 $255 \times 2 + 2 = 512$ ，故在链式存储结构下一个物理块可放 2 个记录及一个物理块地址，而文件 A 共有 598 个记录，故读取 A 的所有记录需读盘次数为 $598/2 = 299$ （次），所以将 A 读到内存至少需读盘 $299 + 5 = 304$ （次）。

(2) 当文件为连续文件时，同样需要 5 次读盘可找到 A，且此时对于定长记录文件可实现直接存取，故一次读盘即可读出 487 记录，所以至少需要 $5 + 1 = 6$ （次）读盘。

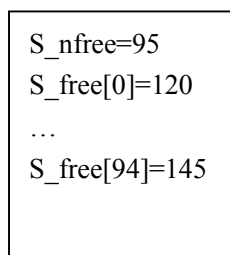
(3) 对查找目录而言可采用降低目录项的大小的方法即索引结点方法。如果一个目录项占 16 字节，则一个盘块可存放 $512/16 = 32$ （个）目录项，与本题一个盘块仅能存放 4 个目录项相比，可使因访问目录而读盘的次数减少 $1/8$ 。对查找文件的记录而言，可用一个或多个盘块来存放该文件的所有盘块号，即用链接索引方法；一个盘块可存放 $512/2 - 1 = 255$ （个）盘块号，剩余的一个地址用来指向下一个存储盘块号（索引块）的磁盘块号。这样，就本题来说，查找目录时需启动 3 次磁盘。文件 A 共有 299 个盘块，则查找文件 A 某一记录时需两次取得所有盘块号，再需最多一次磁盘即可把 A 中任一记录读入内存，所以，查找一记录最多需要 6 次访问盘。

4、答：

(1) 释放四个物理块之后卷资源表如下所示



(2) 分配 6 个空闲块后卷资源表如下所示



5、答：常用的外存空闲存储器空间管理方法：空间表法、位示意图、空闲块连接法、成组连接法。

(1) 空闲表法属于连续分配方法，它为外存上所有空闲区建立一张空闲表，每个空闲区对应一个空闲表项，其中包括序号、该空闲区的第一块号，该区的空闲盘块数等信息，再将所有空闲区按起始盘号递增的次序排列。

(2) 空闲链表法：空闲链表法是将所有空闲盘块或盘区(每个盘区可包含若干个盘块)组织成一条空闲链。当创建文件需要空闲外存空间时，系统从空闲盘块或盘区链表的链首开始依次摘下适当大小的空闲盘块分配给用户；当删除文件时，系统将回收的空闲空间依次插入空闲盘块或盘区链表的末尾。

(3) 位示图法：利用位示图中二进制的的一个位来表示磁盘中一个盘块的使用情况，假设“0”表示对应的盘块空闲；“1”表示被分配。可用 m 行 n 列的位示图中的 $m \times n$ 个位数来表示 $m \times n$ 块盘块的使用情况。

(4) 成组链接法：UNIX 操作系统采用成组链接法管理磁盘空闲块。成组链接法将磁盘空闲块分成若干组，如将每 100 个盘块作为一组，该组空闲块总数和各空闲块块号存入下一组的第一个空闲块中。最后不满 100 块的那组空闲块总数和各空闲块块号记入磁盘区专用管理块的空闲块管理数据结构。由于文件系统要安装时，磁盘专用块拷入内存系统缓存区，所以成组链接法分配和回收操作大部分情况只与内存进行读写，所以速度较快。

6、答：

(1) 根据位示图的位置 (i, j) ，得出盘块的序号 $b = i * 16 + j$ ；用 C 表示柱面号， H 表示磁头号， S 表示扇区号，则有：

$$C = b / (20 * 8)$$

$$H = (b \% (20 * 8)) / 8$$

$$S = (b \% (20 * 8)) \% 8$$

(2) 分配：顺序扫描位示图，找出 1 个其值为“0”的二进制位，利用上述公式将其转换成相应的序号 b ，并修改位示图，置 $(i, j) = 1$ ；

回收：将回收盘块的盘块号换算成位示图中的 i 和 j ，转换公式为：

$$b = C * 20 * 8 + H * 8 + S$$

$i=b/16, j=b\%16$

最后将计算出的 (i,j) 在位示图中置“0”。

7、答：

基本原理：将空闲块分成若干组，假设每 100 个空闲块为一组，将每一组含有的盘块总数和该组所有的盘块号记入其前一组的第一个盘块中，这样可链成一条链，第一组的盘块总数和所有的盘块号记入内存专用块中，作为当前可供分配的空闲盘块号，将最后一组的第一个盘块号用“0”表示，作为空闲盘块链的结束标志。

空闲盘块的分配：首先查看当前组空闲块数，如果大于 1 则减 1，并从专用块中取出一空闲盘块号，将其对应的盘块分配给用户；若专用块中只有一个可分配的盘块号且不是最后一组，须将此盘块号所对应盘块的内容读入专用块中，并把专用块中原来的盘块号对应的盘块分配出去；否则是最后一组，分配不成功。

空闲盘块的回收：若当前组空闲块数不足 100 块，则将回收块的块号记入专用块，空闲块数加 1；当专用块中空闲盘块数为 100 时，便将现有专用块中内容记入新回收的盘块中，然后将盘块数 1 和新回收的盘块号写入专用块。

补充一道非常好的习题：

文件 F 由 200 条记录组成，记录从 1 开始编号。用户打开文件后，欲将内存中的一条记录插入到文件 F 中，作为其第 30 条记录。请回答下列问题，并说明理由。

(1) 若文件系统采用连续分配方式，每个磁盘块存放一条记录，文件 F 存储区域前后均有足够的空闲磁盘空间，则完成上述插入操作最少需要访问多少次磁盘块？F 的文件控制块内容会发生哪些改变？

(2) 若文件系统采用链接分配方式，每个磁盘块存放一条记录和一个链接指针，则完成上述插入操作需要访问多少次磁盘块？若每个磁盘块大小为 1 KB，其中 4 个字节存放链接指针，则该文件系统支持的文件最大长度是多少？

参考答案：本题考察文件系统中记录的插入问题。题目本身比较简单，学生需要区分顺序分配方式和链接分配方式的差别。

(1) 系统采用顺序分配方式时，插入记录需要移动其他的记录块，整个文件共有 200 条记录，要插入新记录作为第 30 条，而存储区前后均有足够的磁盘空间，且要求最少的访问存储块数，则要把文件前 29 条记录前移，若算访盘次数移动一条记录读出和存回磁盘各是一次访盘，29 条记录共访盘 58 次，存回第 30 条记录访盘 1 次，共访盘 59 次。

F 的文件控制区的起始块号和文件长度的内容会因此改变。(2) 文件系统采用链接分配方式时，插入记录并不用移动其他记录，只需找到相应的记录，修改指针即可。插入的记录为其第 30 条记录，那么需要找到文件系统的第 29 块，一共需要访盘 29 次，然后把第 29 块的下块地址部分赋给新块，把新块存回内存会访盘 1 次，然后修改内存中第 29 块的下块地址字段，再存回磁盘，一共访盘 31 次。

4 个字节共 32 位，可以寻址 $2^{32}=4G$ 块存储块，每块的大小为 1KB，即 1024B，其中下块地址部分占 4B，数据部分占 1020B，那么该系统的文件最大长度是 $4G \times 1020B = 4080GB$ 。