第一章 os 引论

1. 设计现代 OS 的主要目标是什么?

方便性，有效性，可扩充性和开放性.

2. OS 的作用可表现为哪几个方面?

a. OS 作为用户与计算机硬件系统之间的接口；

b. OS 作为计算机系统资源的管理者；

c. OS 作为扩充机器.

3. 试说明推动多道批处理系统形成和发展的主要动力是什么?

不断提高计算机资源利用率和系统吞吐量的需要；

4. 何谓脱机 I/O 和联机 I/O?

a. 脱机输入输出方式(Off-Line I/O)是为了解决人机矛盾及 CPU 和 I/O 设备之间速度不匹配而提出的.它减少了 CPU 的空闲等待时间，提高了 I/O 速度.具体内容是将用户程序和数据在一台外围机的控制下，预先从低速输入设备输入到磁带上，当 CPU 需要这些程序和数据时，在直接从磁带机高速输入到内存，从而大大加快了程序的输入过程，减少了 CPU 等待输入的时间，这就是脱机输入技术;当程序运行完毕或告一段落，CPU 需要输出时，无需直接把计算结果送至低速输出设备，而是高速把结果输出到磁带上，然后在外围机的控制下，把磁带上的计算结果由相应的输出设备输出，这就是脱机输出技术.

b. 若这种输入输出操作在主机控制下进行则称之为联机输入输出方式.

5. 试说明推动分时系统形成和发展的主要动力是什么?

用户的需要.即对用户来说，更好的满足了人-机交互，共享主机以及便于用户上机的需求.

6. 试说明实时任务的类型和实时系统的类型.

a. 实时任务的类型按任务执行时是否呈现周期性来划分，分为周期性实时任务和非周期性实时任务；

---根据对截止时间的要求来划分，分为硬实时任务和软实时任务;

b. 通常把要求进行实时控制的系统统称为实时控制系统,把要求对信息进行实时处理的系统成为实时信息处理系统.

7. 实现多道程序应解决哪些问题?

a. 处理机管理问题；

b. 内存管理问题；

c. I/O 设备管理问题；

d. 文件管理问题；

e. 作业管理问题.

8. 试比较单道与多道批处理系统的特点及优缺点.

a. 单道批处理系统是最早出现的一种 OS，它具有自动性，顺序性和单道性的特点；

---多道批处理系统则具有调度性，无序性和多道性的特点；

b. 单道批处理系统是在解决人机矛盾及 CPU 和 I/O 设备之间速度不匹配的矛盾中形成的，旨在提高系统

资源利用率和系统吞吐量，但是仍然不能很好的利用系统资源；

---多道批处理系统是对单道批处理系统的改进，其主要优点是资源利用率高，系统吞吐量大；缺点是平均周转时间长，无交互能力.

9. 实现分时系统的关键问题是什么?应如何解决?

a. 关键问题：及时接收，及时处理;

b. 对于及时接收，只需在系统中设置一多路卡，多路卡作用是使主机能同时接收用户从各个终端上输入的数据；

---对于及时处理，应使所有的用户作业都直接进入内存，在不长的时间内，能使每个作业都运行一次.

10 为什么要引入实时操作系统?

更好地满足实时控制领域和实时信息处理领域的需要.

11 OS 具有哪几大特征?它的最基本特征是什么?

a. 并发(Concurrence),共享(Sharing),虚拟(Virtual),异步性(Asynchronism).

b. 其中最基本特征是并发和共享.

12 内存管理有哪些主要功能?它们的主要任务是什么?

a. 主要功能: 内存分配，内存保护，地址映射和内存扩充等.

b. 内存分配的主要任务是为每道程序分配内存空间，提高存储器利用率，以减少不可用的内存空间，允许正在运行的程序申请附加的内存空间，以适应程序和数据动态增长的需要.

---内存保护的主要任务是确保每道用户程序都在自己的内存空间中运行，互不干扰.

---地址映射的主要任务是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.

---内存扩充的主要任务是借助虚拟存储技术，从逻辑上去扩充内存容量.

13 处理机管理具有哪些功能?它们的主要任务是什么?

a. 进程控制，进程同步，进程通信和调度.

b. 进程控制的主要任务是为作业创建进程，撤销已结束的进程，以及控制进程在运行过程中的状态转换.

---进程同步的主要任务是对诸进程的运行进行调节.

---进程通信的任务是实现在相互合作进程之间的信息交换.

---调度分为作业调度和进程调度.作业调度的基本任务是从后备队列中按照一定的算法，选择出若干个作业，为它们分配必要的资源;而进程调度的任务是从进程的就绪队列中，按照一定的算法选出一新进程，把处理机分配给它，并为它设置运行现场，是进程投入运行.

14 设备管理有哪些主要功能?其主要任务是什么?

a. 主要功能: 缓冲管理，设备分配和设备处理，以及虚拟设备等.

b. 主要任务: 完成用户提出的 I/O 请求，为用户分配 I/O 设备；提高 CPU 和 I/O 设备的利用率；提高I/O 速

度；以及方便用户使用 I/O 设备.

15 文件管理有哪些主要功能?其主要任务是什么?

a. 主要功能: 对文件存储空间的管理，目录管理，文件的读，写管理以及文件的共享和保护.

b. 主要任务: 对用户文件和系统文件进行管理，以方便用户使用，并保证文件的安全性.

16 试在交互性，及时性和可靠性方面，将分时系统与实时系统进行比较.

a. 分时系统是一种通用系统，主要用于运行终端用户程序，因而它具有较强的交互能力；而实时系统虽然也有交互能力，但其交互能力不及前者.

b. 实时信息系统对实用性的要求与分时系统类似，都是以人所能接收的等待时间来确定；而实时控制系统的及时性则是以控制对象所要求的开始截止时间和完成截止时间来确定的.

c. 实时系统对系统的可靠性要求要比分时系统对系统的可靠性要求高.

17 是什么原因使操作系统具有异步性特征?

a. 程序执行结果是不确定的,即程序是不可再现的.

b. 每个程序在何时执行，多个程序间的执行顺序以及完成每道程序所需的时间都是不确定的,即不可预知性.

18 试说明在 MS-DOS 3.X 以前的版本中，其局限性表现在哪几个方面?

a. 在寻址范围上，DOS 只有 1MB，远远不能满足用户需要.

b. DOS 试单用户单任务操作系统，不支持多任务并发执行,与实际应用相矛盾.

19 MS-DOS 由哪几部分组成?每部分的主要功能是什么?

略.

20 为什么 Microsoft 在开发 OS/2 时，选中了 80286 芯片?

设计 OS/2 的主要目标之一是既能充分发挥 80286 处理器的能力，又能运行在 8086 处理器环境下开发的

程序.因为在 80286 内部提供了两种工作方式: 实方式和保护方式，使得 Intel 80286 处理器不仅提供了

多

任务并发执行的硬件支持，而且还能运行所有在 8086 下编写的程序。

21 OS/2 的主要功能是什么?

a. 多任务.

b. 进程管理.

c. 存储器管理.

d. 文件管理.

e. 应用程序接口 API.

f. 表示管理.

22 多处理机 OS 有哪几种模式?各有何优缺点?

a. 2 种模式: 非对称多处理模式(Asymmetric Multiprocessing Model)和对称多处理模式(Symmetric Multiprocesing Model).

b. 前者易于实现，但资源利用率低.

---后者优点是允许多个进程同时运行，缺点是必须小心控制 I/O，以保证能将数据送至适当的处理器，同时还必须注意使各 CPU 的负载平衡.

23 试说明网络 OS 的主要功能.

a. 网络通信;

b. 资源管理;

c. 网络服务;

d. 网络管理;

e. 互操作能力.

24 试比较网络 OS 和分布式 OS.

a. 网络 OS 是基于由一些互联的自主计算机系统组成的计算机网络，以计算机技术和通信技术高度发展为

基础，能实现相互通信和相互合作功能的系统.分布式 OS 是指多个分散的处理单元，经互联网络连接而

形成的系统.

b. 在分布性上，两者都具有分布处理功能，但网络 OS 的控制功能大多集中在某个(些)主机或网络服务器

中，即集中式，而分布式 OS 则是较均匀地分布在系统的各个站点上，是完全分布式的.

---在并行性上，分布式 OS 的任务分配程序可将多个任务分配到多个处理单元上而实现并行，网络 OS 中

通常无任务分配功能,每个用户的任务通常在自己(本地)的计算机上处理.

---在透明性上，两者都具透明性，但网络 OS 指在操作实现上的透明性，而分布式 OS 则在系统内部的细

节上实现了很好的隐藏，即具有物理上的透明性.

---在共享性上，分布式 OS 是比较完全的实现共享，而网络 OS 共享的资源大多是在主机或网络服务器中.

---在健壮性上，分布式系统由于处理和控制功能是分布的，还拥有容错技术实现系统重构，因而具有很强的健壮性；而网络 OS 的控制功能大多集中在主机或服务器中，是系统具有潜在的不可靠性，健壮性差.

第二章

1. 试画出下面条语句的前趋图:

S1: a=5-x; S2: b=a*x; S3: c=4*x; S4: d=b+c; S5: e=d+3.

S1->S2->S4->S5

...../

.....S3

2. 试利用 Bernstein 条件证明上题中的 S2 和 S3 语句是可以并发执行的，而 S3 和 S4 语句是不能并发执行的?

证明:

R(S2)={x,a}, W(S2)={b}, R(S3)={x}, W(S3)={c};

可见，S2 与 S3 的读集与写集两两不相交，S2 与 S3 的读集之间也不相交，因而，他们满足 Bernstein 条件，

S2 与 S3 语句是可以并发执行的.

同理可证 S3 和 S4 不能满足 Bernstein 条件,是不能并发执行的.

3. 程序并发执行为什么会产生间断性?

因为程序在并发执行过程中存在相互制约性.

4. 程序并发执行为何会失去封闭性和可再现性?

因为程序并发执行时，多个程序共享系统中的各种资源，资源状态需要多个程序来改变，即存在资源共

享性使程序失去封闭性；而失去了封闭性导致程序失去可再现性.

5. 在操作系统中为什么要引入进程概念?它会产生什么样的影响?

为了使程序在多道程序环境下能并发执行，并能对并发执行的程序加以控制和描述，而引入了进程概念.

影响: 使程序的并发执行得以实行.

6. 试从动态性，并发性和独立性上比较进程和程序?

a. 动态性是进程最基本的特性，可表现为由创建而产生，由调度而执行，因得不到资源而暂停执行，以及由撤销而消亡，因而进程由一定的生命期；而程序只是一组有序指令的集合，是静态实体.

b. 并发性是进程的重要特征，同时也是 OS 的重要特征.引入进程的目的正是为了使其程序能和其它进程

的程序并发执行，而程序是不能并发执行的.

c. 独立性是指进程实体是一个能独立运行的基本单位，同时也是系统中独立获得资源和独立调度的基本

单位.而对于未建立任何进程的程序，都不能作为一个独立的单位参加运行.

7. 试说明 PCB 的作用?为什么说 PCB 是进程存在的唯一标志?

a. PCB 是进程实体的一部分，是操作系统中最重要的记录型数据结构.PCB 中记录了操作系统所需的用于

描述进程情况及控制进程运行所需的全部信息.因而它的作用是使一个在多道程序环境下不能独立运行

的程序(含数据)，成为一个能独立运行的基本单位，一个能和其它进程并发执行的进程.

b. 在进程的整个生命周期中，系统总是通过其 PCB 对进程进行控制，系统是根据进程的 PCB 而不是任何别

的什么而感知到该进程的存在的，所以说，PCB 是进程存在的唯一标志.

8. 试说明进程在三个基本状态之间转换的典型原因.

a. 处于就绪状态的进程，当进程调度程序为之分配了处理机后，该进程便由就绪状态变为执行状态.

b. 当前进程因发生某事件而无法执行，如访问已被占用的临界资源，就会使进程由执行状态转变为阻塞状态.

c. 当前进程因时间片用完而被暂停执行，该进程便由执行状态转变为就绪状态.

9. 为什么要引入挂起状态?该状态具有哪些性质?

a. 引入挂起状态处于 5 中需要: 终端用户的需要，父进程的需要，操作系统的需要，对换的需要和负荷

调节的需要.

b. 处于挂起状态的进程不能接收处理机调度.

10 在进行进程切换时，所要保存的处理机状态信息主要有哪些?

a. 进程当前暂存信息；
b. 下一条指令地址信息；
c. 进程状态信息；
d. 过程和系统调用参数及调用地址信息.

11 试说明引起进程创建的主要事件.

a. 用户登陆；
b. 作业调度；
c. 提供服务；
d. 应用请求.

12 试说明引起进程撤消的主要事件.

a. 正常结束；
b. 异常结束；
c. 外界干预；

13 在创建一个进程时，需完成的主要工作是什么?

a. 操作系统发现请求创建新进程事件后，调用进程创建原语 Creat()；
b. 申请空白 PCB；
c. 为新进程分配资源；
d. 初始化进程控制块；
e. 将新进程插入就绪队列.

14 在撤消一个进程时，需完成的主要工作是什么?

a. OS 调用进程终止原语；
b. 根据被终止进程的标志符，从 PCB 集合中检索出该进程的 PCB，从中读出该进程的状态；
c. 若被终止进程正处于执行状态，应立即中止该进程的执行，并设置调度标志为真；
d. 若该进程还有子孙进程，还应将其所有子孙进程予以终止；
e. 将该进程所拥有的全部资源，或者归还给其父进程，或者归还给系统；
f. 将被终止进程(它的 PCB)从所在队列(或链表)中移出，等待其它程序来搜集信息.

15 试说明引起进程阻塞或被唤醒的主要事件是什么?

a. 请求系统服务；
b. 启动某种操作；
c. 新数据尚未到达；
d. 无新工作可做.

16 试从调度性，并发性，拥有资源及系统开销几个方面，对进程和线程进行比较.

a. 在引入线程的 OS 中，把线程作为调度和分派的基本单位，而把进程作为资源拥有的基本单位；
b. 在引入线程的 OS 中，不仅进程之间可以并发执行，而且在一个进程中的多个线程之间，亦可并发执行，因而使 OS 具有更好的并发性；
c. 进程始终是拥有资源的一个独立单位，线程自己不拥有系统资源，但它可以访问其隶属进程的资源；
d. 在创建，撤消和切换进程方面，进程的开销远远大于线程的开销.

17 什么是用户级线程和内核级线程?并对它们进行比较.

a. 内核级线程是依赖于内核的，它存在于用户进程和系统进程中，它们的创建，撤消和切换都由内核实现；
---用户级线程仅存在于用户级中，它们的创建，撤消和切换不利用系统调用来实现，因而与内核无关，内核并不知道用户级线程的存在.
b. 内核级线程的调度和切换与进程十分相似，调度方式采用抢占式和非抢占式，调度算法采用时间轮转
法和优先权算法等，当由线程调度选中一个线程后，再将处理器分配给它；而用户级线程通常发生在一个应用程序的诸线程之间，无需终端进入 OS 内核，切换规则也较简单，因而，用户级线程的切换

速度较快.

---用户级线程调用系统调用和调度另一个进程执行时，内核把它们看作是整个进程的行为，内核级线程

调用是以线程为单位，内核把系统调用看作是该线程的行为.

---对于用户级线程调用，进程的执行速度随着所含线程数目的增加而降低，对于内核级线程则相反.

18 在 Solaris OS 中，设置了哪几种线程?轻型线程的作用是什么?

a. 用户级线程，内核级线程和轻型线程；

b. 作用: 由 LWP 实现了在内核与用户级线程之间的隔离，从而使用户级线程与内核无关.

19 在 Solaris OS 中，用户级线程是通过什么方式来访问内核的?

通过 LWP 来访问内核. LWP 可为内核所识别，但不能识别用户级线程，通过建立用户级线程与 LWP 之间的

连接，可以实现用户级线程与内核的通信.

第三章

1. 什么是临界资源和临界区?

a. 一次仅允许一个进程使用的资源成为临界资源.

b. 在每个进程中，访问临界资源的那段程序称为临界区.

2. 为什么进程在进入临界区之前，应先执行"进入区"代码，在退出临界区后又执行"退出区"代码?

为了实现多个进程对临界资源的互斥访问，必须在临界区前面增加一段用于检查欲访问的临界资源是否正被访问的代码，如果未被访问，该进程便可进入临界区对资源进行访问，并设置正被访问标志,如果正被访问，则本进程不能进入临界区，实现这一功能的代码成为"进入区"代码；在退出临界区后，必须执行"退出区"代码，用于恢复未被访问标志.

3. 同步机构应遵循哪些基本准则?为什么?

a. 空闲让进.

b. 忙则等待.

c. 有限等待.

d. 让权等待.

4. 试从物理概念上来说明记录型信号量和 wait 和 signal 操作?

(有待讨论).

5. 你认为整型信号量机制和记录型信号量机制，是否完全遵循了同步机构的四条准则?

a. 在整型信号量机制中，未遵循"让权等待"的准则.

b. 记录型信号量机制完全遵循了同步机构的"空闲让进,忙则等待,有限等待,让权等待"四条准则.

6. 在生产者－消费者问题中，如果缺少了 signal(full)或 signal(empty),对执行结果会有何影响?

生产者－消费者问题可描述如下:

```
var mutex,empty,full: semaphore:=1,n,0;
buffer: array[0,...,n-1] of item;
in,out: integer:=0,0;
begin
parbegin
producer: begin
repeat
.
.
produce an item in nextp;
.
.
wait(empty);
```

```
wait(mutex);
buffer(in):=nextp;
in:=(in+1) mod n;
signal(mutex);
/* ************** */
signal(full);
/* ************** */
until false;
end
consumer: begin
repeat
wait(full);
wait(mutex);
nextc:=buffer(out);
out:=(out+1) mod n;
signal(mutex);
/* ************** */
signal(empty);
/* ************** */
consume the item in nextc;
until false;
end
parend
end
```

可见，生产者可以不断地往缓冲池送消息，如果缓冲池满，就会覆盖原有数据，造成数据混乱.而消费者始终因 wait(full)操作将消费进程直接送入进程链表进行等待，无法访问缓冲池,造成无限等待.

7. 在生产者－消费者问题中，如果将两个 wait 操作即 wait(full)和 wait(mutex)互换位置；或者是将 signal(mutex)与 signal(full)互换位置结果会如何?

```
var mutex,empty,full: semaphore:=1,n,0;
buffer: array[0,...,n-1] of item;
in,out: integer:=0,0;
begin
parbegin
producer: begin
repeat
.
.
produce an item in nextp;
.
.
wait(empty);
wait(mutex);
buffer(in):=nextp;
in:=(in+1) mod n;
/* ***************** */
signal(full);
```

```
signal(mutex);
/* ***************** */
until false;
end
consumer: begin
repeat
/* ***************** */
wait(mutex);
wait(full);
/* ***************** */
nextc:=buffer(out);
out:=(out+1) mod n;
signal(mutex);
signal(empty);
consume the item in nextc;
until false;
end
parend
end
```

a. wait(full)和 wait(mutex)互换位置后，因为 mutex 在这儿是全局变量，执行完 wait(mutex)，则 mutex 赋值为 0，倘若 full 也为 0，则该生产者进程就会转入进程链表进行等待，而生产者进程会因全局变量 mutex 为 0 而进行等待，使 full 始终为 0，这样就形成了死锁.

b. 而 signal(mutex)与 signal(full)互换位置后，从逻辑上来说应该是一样的.

8. 我们为某临界区设置一把锁 W，当 W=1 时，表示关锁；W=0 时，表示锁已打开.试写出开锁原语和关锁原语，并利用它们去实现互斥.

开锁原语:

```
unlock(W):
W=0;
```

关锁原语:

```
lock(W);
if(W==1) do no_op;
W=1;
```

利用开关锁原语实现互斥:

```
var W: semaphore:=0;
begin
parbegin
process :
begin
repeat
lock(W);
critical section
unlock(W);
remainder section
until false;
end
parend
```

9. 试修改下面生产者－消费者问题解法中的错误:

```
producer:
begin
repeat
.
.
producer an item in nextp;
wait(mutex);
wait(full); /* 应为 wait(empty),而且还应该在 wait(mutex)的前面 */
buffer(in):=nextp;
/* 缓冲池数组游标应前移: in:=(in+1) mod n; */
signal(mutex);
/* signal(full); */
until false;
end
consumer:
begin
repeat
wait(mutex);
wait(empty); /* 应为 wait(full),而且还应该在 wait(mutex)的前面 */
nextc:=buffer(out);
out:=out+1; /* 考虑循环，应改为: out:=(out+1) mod n; */
signal(mutex);
/* signal(empty); */
consumer item in nextc;
until false;
end
```

10 试利用记录型信号量写出一个不会出现死锁的哲学家进餐问题的算法.

设初始值为 1 的信号量 c[I]表示 I 号筷子被拿(I=1,2,3,4,...,2n),其中 n 为自然数.

```
send(I):
Begin
if I mod 2==1 then
{
P(c[I]);
P(c[I-1 mod 5]);
Eat;
V(c[I-1 mod 5]);
V(c[I]);
}
else
{
P(c[I-1 mod 5]);
P(c[I]);
Eat;
V(c[I]);
V(c[I-1 mod 5]);
}
End
```

11 在测量控制系统中的数据采集任务，把所采集的数据送一单缓冲区；计算任务从该单缓冲中取出数据进行计算.试写出利用信号量机制实现两者共享单缓冲的同步算法.

```
int mutex=1;
int empty=n;
int full=0;
int in=0;
int out=0;
main()
{
cobegin
send();
obtain();
coend
}
send()
{
while(1)
{
.
.
collect data in nextp;
.
.
wait(empty);
wait(mutex);
buffer(in)=nextp;
in=(in+1) mod n;
signal(mutex);
signal(full);
}
}//send
obtain()
{
while(1)
{
wait(full);
wait(mutex);
nextc:=buffer(out);
out:=(out+1) mod n;
signal(mutex);
signal(empty);
culculate the data in nextc;
}//while
}//obtain
```

12 画图说明管程由哪几部分组成?为什么要引入条件变量?

管程由三部分组成:局部于管程的共享变量说明；对该数据结构进行操作的一组过程；对局部于管程的数据设置初始值的语句. (图见 P80)

因为调用 wait 原语后，使进程等待的原因有多种，为了区别它们，引入了条件变量.

13 如何利用管程来解决生产者－消费者问题?

(见 P82)

14 什么是 AND 信号量?试利用 AND 信号量写出生产者－消费者问题的解法.

为解决并行所带来的死锁问题，在 wait 操作中引入 AND 条件，其基本思想是将进程在整个运行过程中所

需要的所有临界资源，一次性地全部分配给进程，用完后一次性释放.

解决生产者－消费者问题可描述如下:

```
var mutex,empty,full: semaphore:=1,n,0;
buffer: array[0,...,n-1] of item;
in,out: integer:=0,0;
begin
parbegin
producer: begin
repeat
.
.
produce an item in nextp;
.
.
wait(empty);
wait(s1,s2,s3,...,sn); //s1,s2,...,sn 为执行生产者进程除 empty 外其余的条件
wait(mutex);
buffer(in):=nextp;
in:=(in+1) mod n;
signal(mutex);
signal(full);
signal(s1,s2,s3,...,sn);
until false;
end
consumer: begin
repeat
wait(full);
wait(k1,k2,k3,...,kn); //k1,k2,...,kn 为执行消费者进程除 full 外其余的条件
wait(mutex);
nextc:=buffer(out);
out:=(out+1) mod n;
signal(mutex);
signal(empty);
signal(k1,k2,k3,...,kn);
consume the item in nextc;
until false;
end
parend
end
```

15 在单处理机环境下，进程间有哪几种通信方式?

a. 共享存储器系统通信方式;

b. 消息传递系统通信方式;

c. 管道通信方式.

16 试比较进程间的低级通信工具与高级通信工具.

用户用低级通信工具实现进程通信很不方便，因为其效率低，通信对用户不透明，所有的操作都必须由程序员来实现. 而高级通信工具则可弥补这些缺陷，用户可直接利用操作系统所提供的一组通信命令，

高效地传送大量的数据.

17 消息队列通信机制应有哪几方面功能?

略

18 试比较消息队列与管道通信机制.

a. 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共享文件，又称 pipe 文件.

管道通信是属于共享存储器系统的.

b. 消息队列通信机制属于消息传递系统通信机制，存在通信链路，有消息的格式，有若干缓冲队列，采用独特的发送原语和接收原语. (详见 P89－90)

第四章

1. 高级调度与低级调度的主要任务是什么?为什么要引入中级调度?

a. 作业调度又称宏观调度或高级调度，其主要任务是按一定的原则对外存上处于后备状态的作业进行---选择，给选中的作业分配内存，输入输出设备等必要的资源，并建立相应的进程，以使该作业的进---程获得竞争处理机的权利.

b. 进程调度又称微观调度或低级调度，其主要任务是按照某种策略和方法选取一个处于就绪状态的进---程，将处理机分配给它.

c. 为了提高内存利用率和系统吞吐量，引入了中级调度.

2. 在作业调度中需做出哪两个决定?

a. 接纳多少个作业;

b. 接纳哪些作业.

3. 在剥夺调度方式中,有哪些剥夺原则?

a. 时间片原则;

b. 优先权原则;

c. 短作业(进程)优先原则.

4. 在 OS 中引起进程调度的主要因素有哪些?

(有待讨论)

5. 选择调度方式和调度算法时，应遵循的准则是什么?

a. 面向用户的准则有周转时间短，响应时间快，截止时间的保证，以及优先权准则.

b. 面向系统的准则有系统吞吐量高，处理机利用率好，各类资源的平衡利用.

6. 在批处理系统，分时系统和实时系统中，各采用哪几种进程(作业)调度算法?

(有待讨论)

7. 为什么说多级反馈队列能较好地满足各种用户的需要?

a. 对于终端型作业用户，由于终端型作业用户所提交的作业，大都属于交互型作业，系统只要能使这些作业(进程)在第一队列所规定的时间片内完成，便可使终端型作业用户都感到满意.

b. 对于短批处理作业用户，很短的批处理型作业如果仅在第一队列中执行一个时间片即可完成，便可---获得与终端型作业一样的相应时间.对于稍长的作业，通常也只需在第二队列和第三队列中各执行---一个时间片即可完成，其周转时间仍然很短.

c. 对于长批处理作业用户，用户也不必担心其作业长期得不到处理.

8. 在按时间片轮转调度算法中，在确定时间片的大小时，应考虑哪些因素?

a. 系统对相应时间的要求；
b. 就绪队列中进程的数目；
c. 系统的处理能力.

9. 为实现实时调度，对实时系统提出了哪些要求?

a. 要提供必要的调度信息；
b. 在调度方式上要具体情况具体分析；
c. 要具有快速响应外部中断的能力；
d. 快速任务分派.

10 目前常用的调度方式和算法，能否应用到实时系统中?

a. 对于时间片轮转调度算法，是一种常用于分时系统的调度算法;
b. 对于非抢占式优先权调度算法，可用于要求不太严格的实时控制系统中;
c. 对于基于时钟中断抢占的优先权调度算法，有很好的响应效果，可用于大多数的实时系统中;
d. 对于立即抢占(Immediate Preemption)的优先权调度，要求操作系统具有快速响应外部时间的能力.

11 在多处理机系统中，比较有代表性的线程调度方式有哪几种?

a. 自调度方式；
b. 成组调度；
c. 专用处理机分配调度方式.

12 试比较自调度和成组调度?

a. 自调度方式是系统中有一个公共的线程或进程的就绪队列，所有的处理机在空闲时，都可自己从
---该队列中取出一个进程或线程运行;
b. 成组调度是由系统将一组相关的进程或线程，同时分配到一组处理机上运行，进程或线程与处理机
---一一对应;
c. 在一般情况下，成组调度的性能优于自调度，因为自调度存在瓶颈，低效，线程切换频繁等问题，而
---成组调度可减少线程的切换和调度的开销，因而目前得到了广泛的认可.

13 在 OS/2 中采用哪种调度方式和调度算法?

在 OS/2 中采用的是抢占式调度方式，多优先级的抢占式调度算法.

14 何谓死锁?产生死锁的原因和必要条件是什么?

a. 死锁是指多个进程因竞争资源而造成的一种僵局，若无外力作用，这些进程都将永远不能再向前推进;
b. 产生死锁的原因有二，一是竞争资源，二是进程推进顺序非法；
c. 必要条件是：互斥条件，请求和保持条件，不剥夺条件和环路等待条件.

15 在解决死锁问题的几个方法中，哪种方法最容易实现?哪种方法使资源的利用率最高?

a. 解决死锁可归纳为四种方法: 预防死锁，避免死锁，检测死锁和解除死锁；
b. 其中，预防死锁是最容易实现的；
c. 避免死锁使资源的利用率最高.

16 请详细说明可通过哪些途径预防死锁?

a. 摈弃"请求和保持"条件,就是如果系统有足够的资源，便一次性地把进程所需的所有资源分配给它；
b. 摈弃"不剥夺"条件，就是已经保持了资源的进程，当它提出新的资源请求而不能立即得到满足时，
---必须释放它已经保持的所有资源，待以后需要时再重新申请；
c. 摈弃"环路等待"条件，就是将所有资源按类型排序标号，所有进程对资源的请求必须严格按序号递增
---的次序提出.

17 在银行家算法的例子中，如果 P0 发出的请求向量由 Request0(0,2,0)改为 Request0(0,1,0),问系统可否将资源分配给它?

可以.

首先，Request0(0,1,0)<=Need0(7,4,3), Request0(0,1,0)<=Available(2,3,0);

分配后可修改得一资源数据表(表略),进行安全性检查，可以找到一个安全序列{P1,P4,P3,P2,P0}, 或{P1,P4,P3,P0,P2},因此，系统是安全的，可以立即将资源分配给 P0.

第五章

1. 可采用哪几种方式将程序装入内存?它们分别适用于何种场合?

a. 首先由编译程序将用户源代码编译成若干目标模块，再由链接程序将编译后形成的目标模块和所需的
---库函数链接在一起，组成一个装入模块，再由装入程序将装入模块装入内存；
b. 装入模块的方式有: 绝对装入方式，可重定位方式和动态运行时装入方式；
c. 绝对装入方式适用于单道程序环境下；
d. 可重定位方式适用于多道程序环境下；
e. 动态运行时装入方式也适用于多道程序环境下.

2. 何谓静态链接及装入时动态链接和运行时的动态链接?

a. 静态链接是指事先进行链接形成一个完整的装入模块，以后不再拆开的链接方---式；
b. 装入时动态链接是指目标模块在装入内存时，边装入边链接的链接方式；
c. 运行时的动态链接是将某些目标模块的链接推迟到执行时才进行.

3. 在进行程序链接时，应完成哪些工作?

a. 对相对地址进行修改；
b. 变换外部调用符号. -

4. 在动态分区分配方式中，可利用哪些分区分配算法?

a. 首次适应算法；
b. 循环首次适应算法；
c. 最佳适应算法.

5. 在动态分区分配方式中，应如何将各空闲分区链接成空闲分区链?

应在每个分区的起始地址部分，设置一些用于控制分区分配的信息，以及用于链接各分区的前向指针; 在分区尾部则设置一后向指针，通过前，后向指针将所有的分区链接成一个双向链.

6. 为什么要引入动态重定位?如何实现?

a. 为了在程序执行过程中，每当访问指令或数据时，将要访问的程序或数据的逻辑地址转换成物理地---址，引入了动态重定位.
b. 可在系统中增加一个重定位寄存器，用它来装入(存放)程序在内存中的起始地址，程序在执行时，真
---正访问的内存地址是相对地址与重定位寄存器中的地址相加而形成的,从而实现动态重定位.

7. 试用类 Pascal 语言来描述首次适应算法进行内存分配的过程.

(略)

8. 在采用首次适应算法回收内存时，可能出现哪几种情况?应怎样处理这些情况?

a. 回收区与插入点的前一个分区相邻接，此时可将回收区与插入点的前一分区合并，不再为回收分区---分配新表项，而只修改前邻接分区的大小；
b. 回收分区与插入点的后一分区相邻接，此时合并两区，然后用回收区的首址作为新空闲区的首址，大
---小为两者之和；
c. 回收区同时与插入点的前后两个分区邻接，此时将三个分区合并，使用前邻接分区的首址，大小为---三区之和，取消后邻接分区的表项；
d. 回收区没有邻接空闲分区，则应为回收区单独建立一个新表项，填写回收区的首址和大小，并根据---其首址，插入到空闲链中的适当位置.

9. 在系统中引入对换后带有哪些好处?

能将内存中暂时不运行的进程或暂时不用的程序和数据，换到外存上，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和数据换入内存，从而大大地提高了内存的利用率.

10 为实现对换，系统应具备哪几方面功能?

a. 对对换空间的管理；

b. 进程的换出；

c. 进程的换入.

11 在以进程为单位进行对换时，每次是否都将整个进程换出?为什么?

a. 以进程为单位进行对换时，每次都将整个进程换出；

b. 目的为了解决内存紧张的问题，提高内存的利用率.

12 为实现分页存储管理，需要哪些硬件支持?你认为以 Intel 8086,MC68000,

Intel 80286 为芯片的微机，是否适合于实现分页管理?

(有待讨论)

13 请较详细地说明，引入分段存储管理是为了满足用户哪几方面的需要?

a. 方便了编程；

b. 实现了分段共享；

c. 实现了分段保护；

d. 实现了动态链接；

e. 实现了动态增长.

14 在具有快表的段页式存储管理方式中，如何实现地址变换?

首先，必须配置一段表寄存器，在其中存放段表始址和段长 TL. 进行地址变换时，先利用段号 S，与段长 TL

进行比较，若 S<TL，表示未越界，(若 S>=TL,表示段号太大，访问越界，产生越界中断信号)于是利用段表

始址和段号来求出该段对应的段表项在段表中的位置，从中求出该段的页表始址，并利用逻辑地址中的段

内页号 P 来获得对应页的页表项位置，从中读出该页所在的物理块号 b，再用块号 b 和页内地址构成物理地址.

15 为什么说分段系统较之分页系统更易于实现信息共享和保护?

a. 对于分页系统，每个页面是分散存储的，为了实现信息共享和保护，则页面之间需要一一对应起来，为此

---需要建立大量的页表项；

b. 而对于分段系统，每个段都从 0 开始编址，并采用一段连续的地址空间，这样在实现共享和保护时，只需

---为所要共享和保护的程序设置一个段表项，将其中的基址与内存地址一一对应起来即可.

16 分页和分段有何区别?

a. 分页和分段都采用离散分配的方式，且都要通过地址映射机构来实现地址变换，这是它们的共同点；

b. 对于它们的不同点有三，第一，从功能上看，页是信息的物理单位，分页是为实现离散分配方式，以消减

---内存的外零头，提高内存的利用率，即满足系统管理的需要，而不是用户的需要；而段是信息的逻辑单位，

---它含有一组其意义相对完整的信息，目的是为了能更好地满足用户的需要；

c. 页的大小固定且由系统确定，而段的长度却不固定，决定于用户所编写的程序；

d. 分页的作业地址空间是一维的，而分段的作业地址空间是二维的.

17 试全面比较连续分配和离散分配方式.

a. 连续分配是指为一个用户程序分配一个连续的地址空间，包括单一连续分配方式和分区式分配方式，前者

---将内存分为系统区和用户区，系统区供操作系统使用，用户区供用户使用，是最简单的一种存储方式，

---但只能用于单用户单任务的操作系统中；分区式分配方式分为固定分区和动态分区，固定分区是最

简单的

---多道程序的存储管理方式，由于每个分区的大小固定，必然会造成存储空间的浪费；动态分区是根据进程

---的实际需要，动态地为之分配连续的内存空间，常用三种分配算法: 首次适应算法 FF，该法容易留下许多

---难以利用的小空闲分区，加大查找开销；循环首次适应算法，该算法能使内存中的空闲分区分布均匀，但

---会致使缺少大的空闲分区；最佳适应算法，该算法也易留下许多难以利用的小空闲区；

b. 离散分配方式基于将一个进程直接分散地分配到许多不相邻的分区中的思想，分为分页式存储管理，分段

---存储管理和段页式存储管理. 分页式存储管理旨在提高内存利用率，满足系统管理的需要，分段式存储管

---理则旨在满足用户(程序员)的需要，在实现共享和保护方面优于分页式存储管理，而段页式存储管理则是

---将两者结合起来，取长补短，即具有分段系统便于实现，可共享，易于保护，可动态链接等优点，又能像

---分页系统那样很好的解决外部碎片的问题，以及为各个分段可离散分配内存等问题，显然是一种比较有效

---的存储管理方式；

c. 综上可见，连续分配方式和离散分配方式各有各自的特点，应根据实际情况加以改进和利用.

第六章

1. 在请求分页系统中，其页表项中包含那些数据项? 它们的作用是什么?

a. 在请求分页系统中，其页表项中包含的数据项有页号，物理块号，状态位 P，访问字段 A，修改位 M 和

---外存地址；

b. 其中状态位 P 指示该页是否调入内存，供程序访问时参考；

c. 访问字段 A 用于记录本页在一段时间内被访问的次数，或最近已有多长时间未被访问，提供给置换算法

---选择换出页面时参考；

d. 修改位 M 表示该页在调入内存后是否被修改过；

e. 外存地址用于指出该页在外存上的地址，通常是物理块号，供调入该页时使用.

2. 一个计算机系统的虚拟存储器，其最大容量和实际容量分别由什么决定?

a. 最大容量由内存和外存之和决定；

b. 实际容量由内存决定.

3. 虚拟存贮器有那些特征? 其中最本质的特征是什么?

a. 虚拟存储器具有离散性，多次性，对换性和虚拟性的特征；

b. 其中最本质的特征是离散性，在此基础上又形成了多次性和对换性，所表现出来的最重要的特征是

---虚拟性.

4. 实现虚拟存储器要那些硬件支持?

a. 对于为实现请求分页存储管理方式的系统，除了需要一台具有一定容量的内存及外存的计算机外，还

---需要有页表机制，缺页中断机构以及地址变换机构；

b. 对于为实现请求分段存储管理方式的系统，除了需要一台具有一定容量的内存及外存的计算机外，还

---需要有段表机制，缺段中断机构以及地址变换机构；

5. 在实现虚拟存储器时的几个关键技术是什么?

(有待讨论)

6. 在请求分页系统中，页表应包括那些数据项?每项的作用是什么?

(同第一题)

7. 在请求分页系统中，应从何处将所需页面调入内存?

a. 在进行地址变换时，首先去检索快表，试图从中找出所要访问的页，若找到，便修改页表项中的访问

---位，对于写指令，还须将修改位置 1，然后利用页表项中给出的物理块号和页内地址，形成物理地址；

b. 如果在快表中未找到该页的页表项，则应再到内存中去查找页表，再从找到的页表项中的状态位来

---了解该页是否已调入内存，如果该页已调入内存，应将此页的页表项写入快表，当快表已满时，应先

---调出按某种算法所确定的页的页表项，然后再写入该页的页表项；

c. 如果该页尚未调入内存，这时便应产生缺页中断，请求 OS 从外存中把该页调入内存;

d. 外存分为文件区和对换区，若系统有足够的对换区空间，可在进程运行前，将与该进程有关的文件

---拷贝到对换区，需要时从对换区调入；

e. 若系统缺少足够的对换区空间，则凡是不会被修改的文件，可直接从文件区调入，需换出时可不必

---写入外存，但对于可能被修改的部分，在将它们换出时，便须调到对换区，以后需要时再从对换区

---调入.

8. 在请求分页系统中，常采用哪几种页面置换算法?

a. 最佳置换算法；

b. 先进先出算法；

c. 最近最久未使用 LRU 置换算法；

d. Clock 置换算法;

e. 此外，还有最少使用置换算法和页面缓冲算法.

9. 某虚拟存储器的用户空间共有 32 个页面，每页 1KB，主存 16KB. 假定某时刻

---为用户的第 0，1，2，3 页分别分配的物理块号为 5，10，4，7，试将虚拟地址

---0A5C 和 093C 变换为物理地址.

a. 将 0A5C 变换为 2 进制为: 0000,1010,0101,1100,由于页面大小为 1KB 约为 2 的 10 次方，所以 0A5C 的页号

---为 2，对应的物理块号为:4,所以虚拟地址 0A5C 的物理地址为 125C;

b. 将 093C 变换为 2 进制为: 0000,1001,0011,1100，页号也为 2，对应的物理块号也为 4，此时虚拟地址

---093C 的物理地址为 113C.

10 在请求分页系统中，通常采用那种页面分配方式?为什么?

a. 在请求分页系统中，有固定和可变分配两种分配方式；

b. 采用固定分配方式是基于进程的类型(交互型)或根据程序员，系统管理员的建议，为每个进程分配

---一固定页数的内存空间，在整个运行期间不再改变；

c. 采用可变分配方式有全局置换和局部置换两种，前者易于实现，后者效率高.

11 在一个请求分页系统中，采用 LRU 页面置换算法时，假如一个作业的页面走向

---为 4，3，2，1，4，3，5，4，3，2，1，5，当分配给该作业的物理块数 M 分别

---为 3 和 4 时，试计算访问过程中所发生的缺页次数和缺页率?比较所得结果?

a. 当分配给该作业的物理块数 M 为 3 时，所发生的缺页率为 7，缺页率为: 7/12=0.583；

b. 当分配给该作业的物理块数 M 为 4 时，所发生的缺页率为 4，缺页率为: 4/12=0.333.

12 在置换算法中，LRU 和 LFU 哪个更常用?为什么?

a. LRU 与 LFU 置换算法的页面的访问图完全相同，即使用的硬件是相同的；

b. 但是 LFU 并不能真正访问反映出页面的使用情况.

13 实现 LRU 算法所需的硬件支持是什么?

a. 寄存器，用于记录某进程在内存中各页的使用情况；

b. 栈，用于保存当前使用的各个页面的页面号.

14 试说明改进型 Clock 置换算法的基本原理.

a. 因为对于修改过的页面在换出时所付出的开销将比未被修改过的页面的开销大，所以在改进型 Clock

---算法中，出了须考虑到页面的使用情况外，还须再增加一个置换代价这一因素；

b. 在选择页面作为淘汰页面时，把同时满足未使用过和未被修改作为首选淘汰页面.

15 什么是抖动? 产生抖动的原因是什么?

a. 抖动(Thrashing)就是指当内存中已无空闲空间而又发生缺页中断时，需要从内存中调出一页程序或

---数据送磁盘的对换区中，如果算法不适当，刚被换出的页很快被访问，需重新调入，因此需再选一页

---调出，而此时被换出的页很快又要被访问，因而又需将它调入，如此频繁更换页面，以致花费大量的

---时间，我们称这种现象为"抖动";

b. 产生抖动的原因是由于 CPU 的利用率和多道程序度的对立统一矛盾关系引起的，为了提高 CPU 利用率，

---可提高多道程序度，但单纯提高多道程序度又会造成缺页率的急剧上升，导致 CPU 的利用率下降，而

---系统的调度程序又会为了提高 CPU 利用率而继续提高多道程序度，形成恶性循环，我们称这时的进程

---是处于"抖动"状态.

16 试说明请求分段系统中的缺页中断处理过程?

(见 P185 图 6-12)

17 如何实现分段共享?

a. 可在每个进程的段表中，用相应的表项来指向共享段在内存中起始地址；

b. 配置相应的数据结构作为共享段表，可在段表项中设置共享进程计数 Count，每调用一次该共享段,

---Count 指增 1，每当一个进程释放一个共享段时，Count 执行减 1 操作，若减为 0，则由系统回收该共享

---段的物理内存，以及取消在共享段表中该段所对应的表项；

c. 对于一个共享段，应给不同的进程以不同的存取权限；

d. 不同的进程可以使用不同的段号去共享该段.

18 Intel 80386 芯片可支持哪几种方式的存储管理?

a. 不分段也不分页的存储管理方式；

b. 分页不分段的存储管理方式；

c. 分段不分页的存储管理方式；

d. 分段分页存储管理方式.

19 试说明 80386 的分段地址变换机构的工作原理.

a. 采用段寄存器和虚地址结构；

b. 在分段部件中，地址变换是将逻辑地址变换为线性地址，然后送分页部件中.(具体见 P191)

20 试说明 80386 的两级分页地址变换机构的原理.

(见 P193)

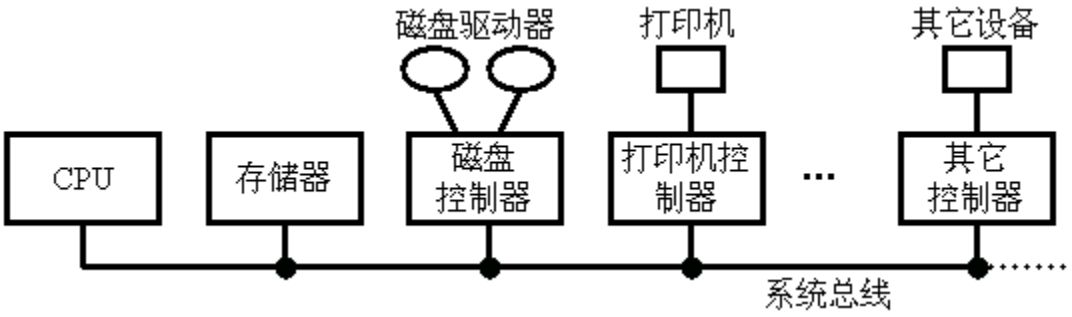
21 可通过哪些途径来提高内存利用率?

(有待讨论,该题可以看成是对本章的本质内容的全面概括和总结)

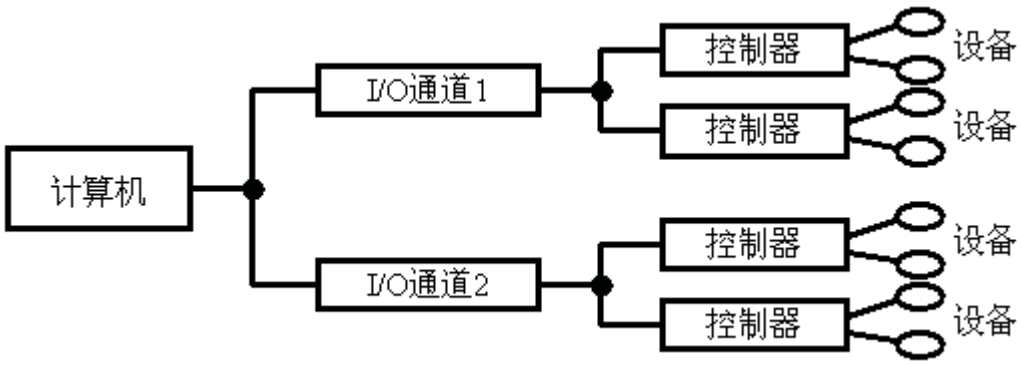
第七章

1. 试画出微机和主机中常采用的 I/O 系统结构图。

微机中常采用的 I/O 系统结构图为：

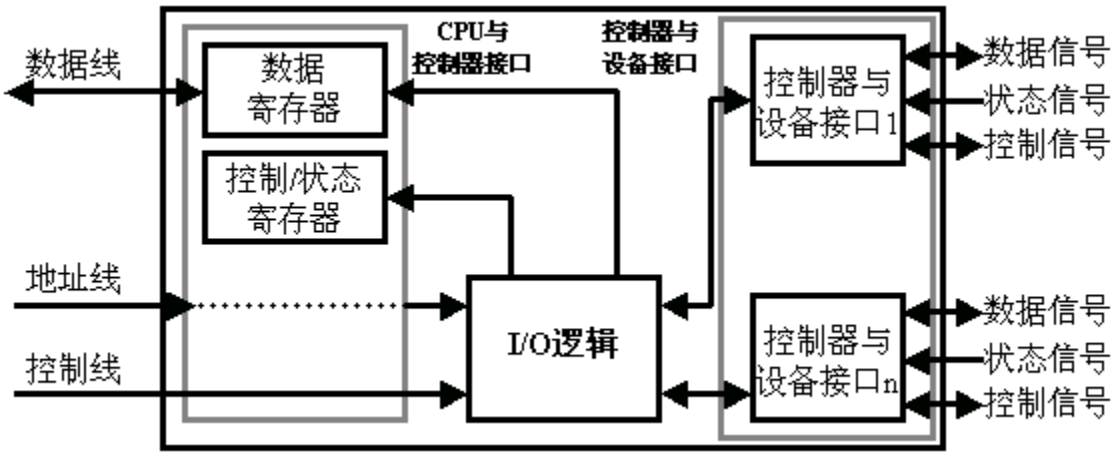


主机中常采用的 I/O 系统结构图为：



2. 试说明设备控制器的构成。

设备控制器的构成如图所示：



由上图可见，设备控制器由以下三部分组成：（1）设备控制器与处理机的接口，该接口用于实现 CPU 与设备控制器之间的通信，提供有三类信号线：数据线、地址线和控制线。（2）设备控制器与设备的接口，可以有一个或多个接口，且每个接口连接一台设备。每个接口都存在数据、控制和状态三种类型的信号。（3）I/O 逻辑，用于实现对设备的控制。其通过一组控制线与处理机交互，处理机利用该逻辑向控制器发送 I/O 命令，I/O 逻辑对收到的命令进行译码。

3. 为了实现 CPU 与设备控制器之间的通信，设备控制器应具有哪些功能？

为了实现 CPU 与设备控制器之间的通信，设备控制器应具有如下功能：（1）接受和识别命令。CPU 可以向控制器发送多种不同的命令，设备控制器应能接收并识别这些命令。设置控制寄存器来存放

所接收的命令和参数。（2）数据交换，指实现 CPU 与控制器之间、控制器与设备之间的数据交换。设置数据寄存器来存放有关数据。（3）设备状态的了解和报告。控制器记录下所连接设备的状态以供 CPU 了解。为此，要在控制器中设置一状态寄存器，用其中的每一位反映设备的某一状态。（4）地址识别。配置地址译码器以便于正确识别设备地址。

4. 分别就字节多路通道、数据选择通道和数组多路通道进行解释。

① 字节多路通道含有许多非分配型子通道分别连接在低、中速 I/O 设备上，子通道按时间片轮转方式共享主通道，按字节方式进行数据传送。具体而言，当第一个子通道控制其 I/O 设备完成一个字节的交换后，便立即腾出字节多路通道（主通道），让给第二个子通道使用；当第二个子通道也交换完一个字节后，又依样把主通道让给第三个子通道使用，以此类推。转轮一周后，重又返回由第一个子通道去使用主通道。② 数组选择通道只含有一个分配型子通道，一段时间内只能执行一道通道程序、控制一台设备按数组方式进行数据传送。通道被某台设备占用后，便一直处于独占状态，直至设备数据传输完毕释放该通道，故而通道利用率较低，主要用于连接多台高速设备。③数组多路通道是将数组选择通道传输速率高和字节多路通道能使各子通道分时并行操作的优点相结合而形成的一种新通道。其含有多个非分配型子通道分别连接在高、中速 I/O 设备上，子通道按时间片轮转方式共享主通道，按数组方式进行数据传送，因而既具有很高的数据传输速率，又能获得令人满意的通道利用率。

5. 如何解决因通道不足而产生的瓶颈问题？

解决因通道不足而产生的瓶颈问题的最有效方法是增加设备到主机间的通路而不是增加通道。换言之，就是把一个设备连接到多个控制器上，而一个控制器又连接到多个通道上。这种多通路方式不仅可以解决该瓶颈问题，而且能够提高系统的可靠性，也即不会因为个别通道或控制器的故障而使设备与存储器之间无法建立通路进行数据传输。

6. 试说明 I/O 控制发展的主要推动因素是什么？

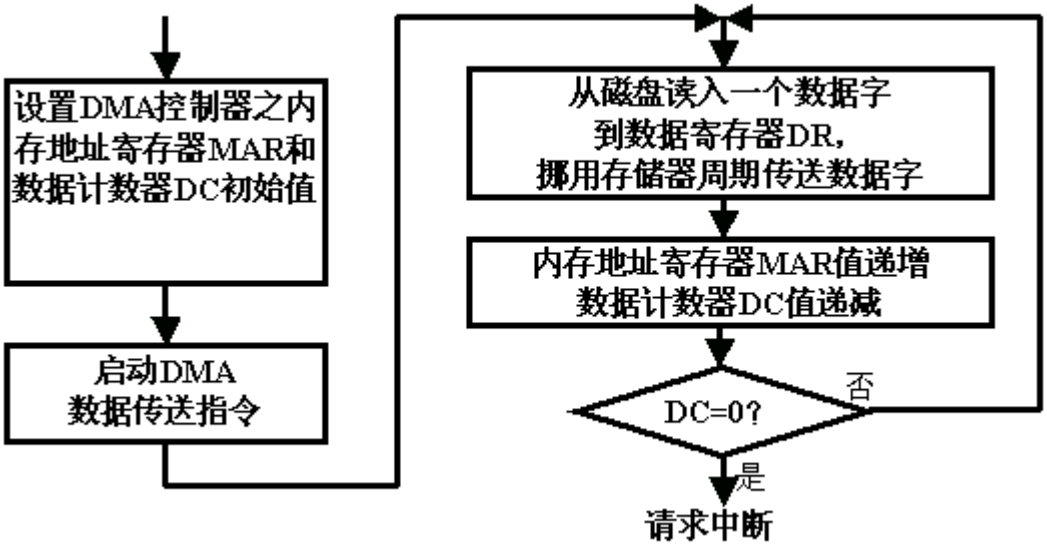
推动 I/O 控制发展的主要动力在于尽量减少主机对 I/O 控制的干预，把主机从繁杂的 I/O 控制事务中解脱出来，以有更多的时间和精力去完成其数据处理任务。同时，中断机制在计算机系统中的引入、DMA 控制器的出现和通道研制的成功使 I/O 控制的发展具备了技术支持和成为可能。

7. 有哪几种 I/O 控制方式？

有四种 I/O 控制方式，即程序 I/O 控制方式、中断驱动 I/O 控制方式、直接存储器访问 DMA 控制方式及 I/O 通道控制方式。

8. 试说明 DMA 的工作流程。

以从磁盘读入数据为例来说明 DMA 方式的工作流程：当 CPU 要从磁盘读入一数据块时，便向磁盘控制器发送一条读命令，该命令被送入 DMA 控制器的命令寄存器 CR 中。同时，还需发送本次要将数据读入的内存起始目标地址，该地址被送入 DMA 控制器的内存地址寄存器 MAR 中；本次要读的字（节）数则送至 DMA 控制器的数据计数器 DC 中。另外，还需将磁盘中数据读取的源地址直接送到 DMA 控制器的 I/O 控制逻辑上。然后，启动 DMA 控制器进行数据传送。此后，CPU 便可去处理其它任务，而整个的数据传送便由 DMA 控制器负责控制。当 DMA 控制器已从磁盘中读入一个字（节）的数据，并送入 DMA 控制器的数据寄存器 DR 后，再挪用一个存储器周期，将该字（节）传送到 MAR 所指示的内存单元中。接着，便对 MAR 内容加 1 和将 DC 内容减 1。若 DC 内容减 1 后不为 0，表示传送未完，便准备再传送下一个字（节），否则，由 DMA 控制器发出中断请求。参图所示：



9. 引入缓冲的主要原因是什么？

操作系统引入缓冲机制的主要原因可归结为以下几点：（1）缓和 CPU 与 I/O 设备间速度不匹配的矛盾；（2）减少对 CPU 的中断频率，放宽对中断响应时间的限制；（3）提高 CPU 与 I/O 设备之间的并行性。

10. 为什么在单缓冲情况下，系统对一块数据的处理时间为 max(C, T)+M？

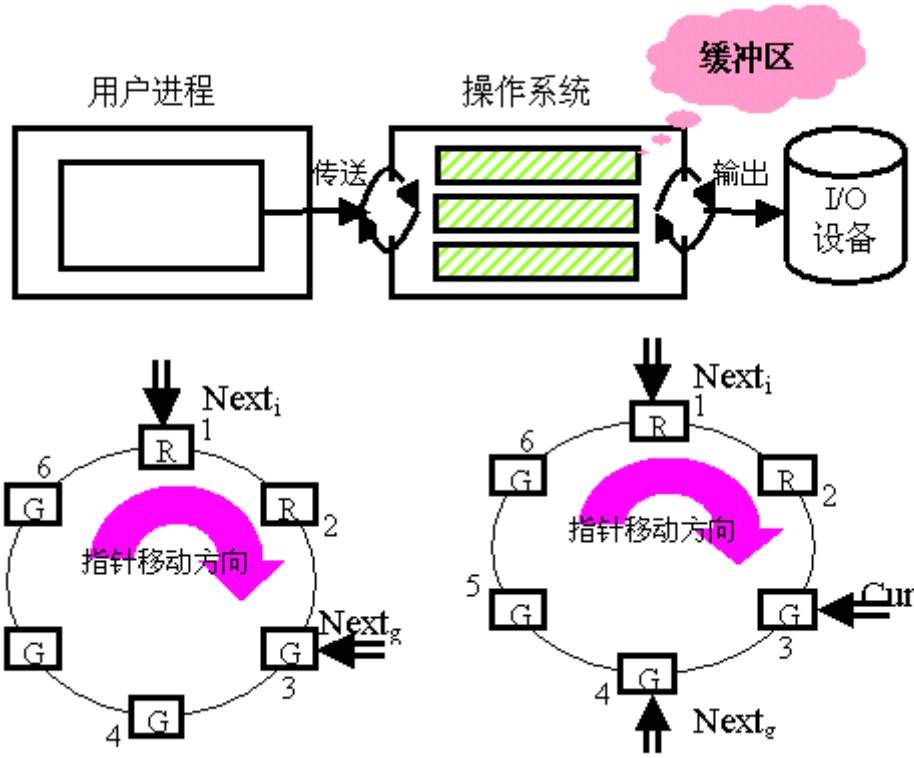
在块设备输入时，先从磁盘把一块数据输入到缓冲区，耗时为 T；然后由操作系统将缓冲区数据传送给用户区，耗时为 M；接下来便由 CPU 对这一块数据进行计算，耗时为 C。在单缓冲情况下，磁盘把数据输入到缓冲区的操作和 CPU 对数据的计算过程可以并行展开，所以系统对每一整块数据的处理时间为 max(C, T) + M。

11. 为什么在双缓冲情况下，系统对一块数据的处理时间为 max(C, T)

该方式又称缓冲对换方式。写入者花费时间 T 将数据写满一个缓冲区后再写另一个缓冲区；读出者花费时间 M 将一个缓冲区数据送到用户区后再传送另一个缓冲区数据，运算者读出用户区进行处理。由于将数据从缓冲区传送到用户区操作必须与读用户区数据进行处理串行进行，而且它们又可以与从外存传送数据填满缓冲区的操作并行。因此耗时大约为 max(C+M,T)。考虑到 M 是内存中数据块的“搬家”耗时，非常短暂可以省略，因此近似地认为是：max(C,T)。

12. 试绘图说明把多缓冲用于输出时的情况。

把多缓冲用于输出时的情况如图所示：



13. 试说明收容输入工作缓冲区和提取输出工作缓冲区的工作情况。

① 收容输入工作缓冲区的工作情况为：在输入进程需要输入数据时，调用 GetBuf(EmptyQueue)过程，从 EmptyQueue 队列的队首摘下一个空缓冲区，把它作为收容输入工作缓冲区 Hin。然后，把数据输入其中，装满后再调用 PutBuf(InputQueue, Hin)过程，将该缓冲区挂在输入队列 InputQueue 的队尾。

② 提取输出工作缓冲区的工作情况为：当要输出数据时，调用 GetBuf(OutputQueue)过程，从输出队列的队首取得一装满输出数据的缓冲区作为提取输出的工作缓冲区 Sout。在数据提取完后，再调用 PutBuf(EmptyQueue, Sout)过程，将该缓冲区挂到空缓冲队列 EmptyQueue 的队尾。

14. 什么是安全分配方式和不安全分配方式？

① 所谓安全分配方式，是指每当进程发出 I/O 请求后，便进入阻塞状态，直到其 I/O 操作完成时才被唤醒。在采用这种分配策略时，一旦进程已经获得某种设备（资源）后便阻塞，使它不可能再请求任何资源，而在它运行时又不保持任何资源。因此，这种分配方式已经摒弃了造成死锁的四个必要条件之一的“请求和保持”条件，所以分配是安全的。其缺点是进程进展缓慢，即 CPU 与 I/O 设备是串行工作的。② 所谓不安全分配方式，是指进程发出 I/O 请求后仍继续执行，需要时又可发出第二个 I/O 请求、第三个 I/O 请求。仅当进程所请求的设备已被另一个进程占有时，进程才进入阻塞状态。其优点是一个进程可同时操作多个设备，从而使进程推进迅速。而缺点是分配不安全，因为它可能具有“请求和保持”条件，所以可能造成死锁。因此，在设备分配程序中还需增加一个功能，用于对本次的设备分配是否会发生死锁进行安全性计算，仅当计算结果说明分配是安全的情况下才进行分配。

15. 为什么要引入设备独立性？如何实现设备独立性？

在现代操作系统中，为了提高系统的可适应性和可扩展性，都毫无例外地实现了设备独立性，也即设备无关性。其基本含义是，应用程序独立于具体使用的物理设备，即应用程序以逻辑设备名称来请求使用某类设备。进一步说，在实现了设备独立性的功能后，可带来两方面的好处：（1）设备分配时的灵活性；（2）易于实现 I/O 重定向(指用于 I/O 操作的设备可以更换即重定向，而不必改变应用程序)。

为了实现设备的独立性，应引入逻辑设备和物理设备两个概念。在应用程序中，使用逻辑设备名称来请求使用某类设备；而系统执行时，是使用物理设备名称。鉴于驱动程序是一个与硬件（或设备）紧密相关的软件，必须在驱动程序之上设置一层软件，称为设备独立性软件，以执行所有设备的公有操作、完成逻辑设备名到物理设备名的转换（为此应设置一张逻辑设备表）并向用户层（或文件

层）软件提供统一接口，从而实现设备的独立性。

16. 在考虑到设备的独立性时，应如何分配独占设备？

在考虑到设备的独立性时，应按如下步骤来分配独占设备：（1）进程以逻辑设备名提出 I/O 请求。（2）根据逻辑设备表相应表项获得 I/O 请求的逻辑设备对应类型的物理设备在系统设备表中的指针。（3）从指针所指位置起顺序检索系统设备表，直到找到一个属于对应 I/O 请求所用类型、空闲可用且基于设备分配安全性算法验证为安全分配的设备的设备控制表，将对应设备分配给请求进程；如果未找到安全可用的空闲设备，则把请求进程的进程控制块挂到相应类型设备的等待队列上等待唤醒和分配。（4）系统把设备分配给 I/O 请求进程后，再到该设备的设备控制表中找出与其相连接的控制器的控制器控制表，根据其状态字段判断该控制器是否忙碌，若忙则把请求进程的进程控制块挂到该控制器的等待队列上；否则将该控制器分配给进程。（5）系统把控制器分配给 I/O 请求进程后，再到该控制器的控制器控制表中找出与其相连接的通道的通道控制表，根据其状态字段判断该通道是否忙碌，若忙则把请求进程的进程控制块挂到该通道的等待队列上；否则将该通道分配给进程。（6）只有在设备、控制器和通道三者都分配成功时，这次的设备分配才算成功，然后便可启动设备进行数据传送。

17. 什么是虚拟设备？其实现所依赖的关键技术有哪些？

通过虚拟技术可将一台独占设备变换成若干台逻辑设备，供若干个用户（进程）同时使用，通常把这种经过虚拟技术处理后的设备称为虚拟设备。其实现所依赖的关键技术是 SPOOLING 技术。

18. 试说明 SPOOLING 系统的组成。

SPOOLing 系统是对脱机 I/O 工作的模拟，其必须有高速随机外存（通常采用磁盘）的支持。SPOOLING 系统主要有以下四个部分：（1）输入井和输出井，为磁盘上开辟的两大存储空间，分别模拟脱机输入/出时的磁盘，并用于收容 I/O 设备输入的数据和用户程序的输出数据；（2）输入缓冲区和输出缓冲区，在内存中开辟，分别用于暂存由输入设备和输出井送来的数据；（3）输入进程 SPi 和输出进程 SPo，分别模拟脱机输入/出时的外围控制机，用于控制 I/O 过程；（4）I/O 请求队列，由系统为各个 I/O 请求进程建立的 I/O 请求表构成的队列。

19. 在实现后台打印时，SPOOLING 系统应为请求 I/O 的进程提供哪些服务？

在实现后台打印时，SPOOLING 系统应为请求 I/O 的进程提供以下服务：（1）由输出进程在输出井中为之申请一空闲盘块区，并将要打印的数据送入其中；（2）输出进程再为用户进程申请一张空白的用户打印表，并将用户的打印要求填入其中，再将该表挂到请求打印队列上。（3）一旦打印机空闲，输出进程便从请求打印队列的队首取出一张请求打印表，根据表中的要求将要打印的数据从输出井传送到内存缓冲区，再由打印机进行打印。

20. 试说明设备驱动程序具有哪些特点？

设备驱动程序具有如下特点：（1）驱动程序主要是在请求 I/O 的进程与设备控制器之间的一个通信程序；（2）驱动程序与 I/O 设备的特性紧密相关；（3）驱动程序与 I/O 控制方式紧密相关；（4）驱动程序与硬件紧密相关，因而其中的一部分程序必须用汇编语言书写，且基本部分往往已被固化在 ROM 中。

21. 试说明设备驱动程序应具有哪些功能？

设备驱动程序的主要功能包括：（1）将接收到的抽象要求转为具体要求；（2）检查用户 I/O 请求的合法性，了解 I/O 设备的状态，传递有关参数，设置设备的工作方式；（3）发出 I/O 命令，启动分配到的 I/O 设备，完成指定的 I/O 操作；（4）及时响应由控制器或通道发来的中断请求，并根据其中断类型调用相应的中断处理程序进行处理；（5）对于设置有通道的计算机系统，驱动程序还应该能够根据用户的 I/O 请求，自动地构成通道程序。

22. 设备驱动程序通常要完成哪些工作？

设备驱动程序通常要完成以下工作：（1）将抽象要求转换为具体要求；（2）检查 I/O 请求的合法性；（3）读出和检查设备的状态；（4）传送必要的参数；（5）设置工作方式；（6）启动 I/O 设备。

23. 设备中断处理程序通常需完成哪些工作？

设备中断处理程序通常需完成如下工作：（1）唤醒被阻塞的驱动程序进程；（2）保护被中断进程的 CPU 环境；（3）分析中断原因、转入相应的设备中断处理程序；（4）进行中断处理；（5）恢复被中断进程

的现场。

第八章

1. 分别就数据项、记录和文件的概念进行解释。

数据项可分为基本数据项和组合数据项。基本数据项是用于描述一个对象某种属性的字符集，是数据组织中可以命名的最小逻辑数据单位，又称为原子数据、数据元素或字段，其具有数据名、数据类型及数据值三个特性。组合数据项则由若干数据项构成。记录是一组相关数据项的集合，用于描述一个对象某方面的属性。文件是具有文件名的一组相关信息的集合。

2. 按文件的物理结构，可将文件分为哪几类？

按文件的物理结构，可将文件分为三类：（1）顺序文件，指把逻辑文件中的记录顺序地存储到连续的物理盘块中；（2）链接文件，指文件中的各个记录可以存放在不相邻的各个物理块中，但通过物理块中的链接指针，将它们链接成一个链表；（3）索引文件，指文件中的各个记录可以存放在不相邻的各个物理块中，但通过为每个文件建立一张索引表来实现记录和物理块之间的映射关系。

3. 文件系统的模型可分为三层，试说明其每一层所包含的基本内容。

答： 文件系统模型如图所示：

<table>
<tr><td colspan="2">文件系统接口（命令/图形化用户接口、程序接口）</td></tr>
<tr><td rowspan="4">对对象操纵和管理的软件集合</td><td>逻辑文件系统层</td></tr>
<tr><td>基本I/O管理程序(文件组织模块)层</td></tr>
<tr><td>基本文件系统(物理I/O层)</td></tr>
<tr><td>I/O控制层(设备驱动程序)</td></tr>
<tr><td colspan="2">对象及其属性说明（文件、目录、磁盘）</td></tr>
</table>

（1）最低层为对象及其属性说明，主要包括文件、目录、磁盘存储空间等三类对象。（2）最高层是文件系统提供给用户的接口，分为命令接口、程序接口和图形化用户接口等三种类型。（3）中间层是对对象进行操纵和管理的软件集合，是文件系统的核心部分，拥有文件存储空间管理、文件目录管理、地址映射、文件读写管理及文件共享与保护等诸多功能。具体又可分为四个子层：①I/O 控制层（又称为设备驱动程序层），主要由磁盘驱动程序和磁带驱动程序组成，负责启动 I/O 设备和对设备发来的中断信号进行处理；②基本文件系统层（又称为物理 I/O 层），主要用于处理内存与磁盘或磁带机系统之间数据块的交换，通过向 I/O 控制层发送通用指令及读写的物理盘块号与缓冲区号等 I/O 参数来完成；③基本 I/O 管理程序层（即文件组织模块层），负责完成与磁盘 I/O 有关的大量事务，包括文件所在设备的选定、文件逻辑块号到物理块号的转换、空闲盘块的管理及 I/O 缓冲的指定等；④逻辑文件系统层，负责所读写的文件逻辑块号的确定、目录项的创建与修改、文件与记录的保护等。

文件系统接口
对对象操纵和管理的软件集合 逻辑文件系统
基本 I/O 管理程序（文件组织模块）
基本文件系统（物理 I/O 层）
I/O 控制层（设备驱动程序）
对象及其属性说明

4. 对于一个较完善的文件系统，应具备哪些功能？

对于一个较完善的文件系统，应具备一系列的功能，包括对文件存储空间的管理、目录管理、文件的读写管理以及文件的共享与保护等。其中，有些功能对用户是透明的，就呈现在用户面前的功能来说，可通过用户对文件所能施加的操作来表现。对文件的操作可分为两大类：一类是对文件自身的操作，包括文件的创建、删除、读、写、截断及文件读/写位置的设置；一类是对记录的操作，包括记录的遍

历（即检索所有记录）、单个记录的检索以及记录的插入、修改和删除。

5. 什么是文件的逻辑结构？什么是文件的物理结构？

文件的逻辑结构是指从用户的观点出发所观察到的文件组织形式，也就是用户可以直接处理的数据及其结构，它独立于物理特性；而文件的物理结构则是指文件在外存上的存储组织形式，与存储介质的存储性能有关。

6. 你认为内存管理和外存管理有哪些相同点和不同点？

内存管理和外存管理均追求存储空间利用率的提高，都具有存储空间的分配与回收、地址映射、共享与保护等功能。但二者的目的和任务不同，因而技术侧重点也有所不同。具体而言，内存管理着眼于为多道程序的运行提供良好的环境，以进程作为分配对象，并要求能从逻辑上扩充内存；而外存管理则着眼于为每个文件分配必要的外存空间，并能有助于提高文件系统的工作速度特别是文件的访问速度。

7. 如何提高对变长记录顺序文件的检索速度？

为了提高对变长记录顺序文件的检索速度，可为其建立一张索引表，以主文件中每条记录的长度及指向对应记录的指针（即该记录在逻辑地址空间的首址）作为相应每个表项的内容。由于索引表本身是一个定长记录的顺序文件，若将其按记录键排序，则可以实现对主文件的方便快速的直接存取。需要指出的是，如果文件较大，应通过建立分组多级索引以进一步提高检索效率。

8. 试说明关于索引文件和索引顺序文件的检索方法。

答：①对索引文件进行检索时，首先根据用户（程序）提供的关键字，并利用折半查找法检索索引表，从中找到相应的表项；再利用该表项中给出的指向记录的指针值，去访问对应的记录。②对索引顺序文件进行检索时，首先利用用户（程序）提供的关键字以及某种查找方法，去检索索引表，找到该记录所在记录组中的第一条记录的表项，从中得到该记录组第一个记录在主文件中的位置；然后再利用顺序查找法去查找主文件，从而找到所要求的记录。

索引文件的检索：首先是根据用户（程序）提供的关键字，并利用折半查找法，去检索索引表，从中找到相应的项，再利用该表项中给出的指向记录的指针值，去访问所需的记录。

索引顺序文件检索：首先利用用户（程序）提供的关键字以及某种查找方法，去检索索引表，找到该记录所在记录组中第一个记录的表项，从中得到该记录组第一个记录在主文件中的位置；然后，再利用顺序查找法去查找主文件，从中找到所要求的记录。

9. 试从检索速度和存储费用两方面对索引文件和索引顺序文件进行比较。

假设主文件拥有 N 条记录。对于索引文件，主文件的每条记录均需配置一个索引项，故存储开销为 N；而为检索到具有指定关键字的记录，平均需要查找 N/2 条记录。对于索引顺序文件，应为每个记录分组配置一个索引项，故存储开销为 N1/2；而为检索到具有指定关键字的记录，平均需要查找 N 1/ 2 条记录。对于两级索引顺序文件，存储开销为 N2/3+N1/3；而为检索到具有指定关键字的记录，平均需要查找 1.5N1/3 条记录。

10. 目录管理主要有哪些要求？

答：对文件目录的管理有以下要求：

a) 实现“按名存取”
b) 提高对目录的检索速度
c) 文件共享
d) 允许文件重名

11. 采用单级目录能否满足目录管理的主要要求？

采用单级目录只能实现目录管理的基本功能（即文件的按名存取），而对于其它三项要求则不能满足。

12. 目前广泛采用的目录结构形式是什么？它有什么优点？

目前广泛采用的目录结构形式是树型目录结构，其具有检索效率高、允许重名、便于实现文件共享等一系列优点。

13. Hash 检索法有何优点？有何局限性？

又称杂凑结构或散列结构。这种结构只适用于定长记录文件和按记录随机查找的访问方式。
Hash 结构的思想是通过计算来确定一个记录在存储设备上的存储位置，依次先后存入的两个记录在物理设备上不一定相邻。按 Hash 结构组织文件的两个关键问题是：
定义一个杂凑函数；
解决冲突；

14. 在 Hash 检索法中，如何解决“冲突”问题？

15. 解释关于树型目录结构采用线性检索法的检索过程。

假设用户给定的文件路径名为/Level1/Level2/…/Leveln/datafile，则关于树型目录结构采用线性检索法检索该文件的基本过程为：①读入第一个文件分量名 Level1，用它与根目录文件（或当前目录文件）中各个目录项的文件名顺序地进行比较，从中找出匹配者，并得到匹配项的索引结点号，再从对应索引结点中获知 Level1 目录文件所在的盘块号，将相应盘块读入内存。②对于 2~n，循环执行以下步骤，以检索各级目录文件：读入第 i 个文件分量名 Leveli，用它与最新调入内存的当前目录文件中各个目录项的文件名顺序地进行比较，从中找出匹配者，并得到匹配项的索引结点号，再从对应索引结点中获知 Leveli 目录文件所在的盘块号，将相应盘块读入内存。③读入最后一个文件分量名即 datafile，用它与第 n 级目录文件中各个目录项的文件名进行比较，从而得到该文件对应的索引结点号，进而找到该文件物理地址，目录查找操作成功结束。如果在上述查找过程中，发现任何一个文件分量名未能找到，则停止查找并返回“文件未找到”的出错信息。

16. 基于索引结点的共享方式有哪些优缺点？

就基于索引结点的共享方式而言，其优点在于“建立新的共享链接，并不改变文件拥有者的关系，仅把索引结点共享计数器加 1，所以系统可方便获悉由多少个目录项指向该文件”。同时，该方式也存在所谓“悬空指针”的问题和缺点。具体而言，文件拥有者不能删除自己的文件，否则将留下指向该结点的悬空指针，造成该结点再分配时，系统出错；为此，拥有者只能清除自己的目录项，且要为其它共享者无端付费，直至其它共有者清除该文件?

17. 基于符号链的文件共享方式有哪些优缺点？

就基于符号链的文件共享方式来说，只有文件主才拥有指向其索引结点的指针，而共享该文件的其它用户只有该文件的路径名且没有指向索引结点的指针，所以也就不会发生在文件主删除共享文件后留下所谓“悬空指针”的问题。当文件拥有者把一个共享文件删除后，其它用户试图通过符号链来访问一个被删除的共享文件时将因系统找不到该文件而使访问失败，于是将符号链删除，此时不会有任何其它负面效应。当然，这种方式也存在自己的问题。在其它用户访问共享文件时，系统是根据给定的文件路径名，逐个分量地去查找目录，直至找到该文件的索引结点。因此，在访问共享文件时要多次读盘，使每次访问文件的系统开销加大，且增加了启动磁盘的频率。此外，要为每个共享用户建立一条符号链，而该链实际上是一个文件，尽管该文件非常简单，却仍需为之配置一个索引结点，故而也要消耗一定的磁盘空间。需要指出的是，本共享方式还有一个特殊的优点，即它能够用于链接（通过计算机网络）世界上任何地方的机器中的文件，此时只需提供该文件所在机器的网络地址以及在该机器中的文件路径。

18. 什么是保护域？进程与保护域之间存在着什么动态联系？

保护域规定了进程所能访问的一组（硬件或软件）对象以及相应的操作类型（即访问权）。进程与保护域之间的动态联系是指进程的可用资源集在其整个生命周期中是变化的；也就是说，进程运行在不同的阶段时，需要从一个保护域切换到另外一个保护域。

19. 如何利用拷贝权来扩散某种访问权？

如果域 i 具有关于对象 j 的某访问权 access(i,j)的拷贝权，则运行在域 i 的进程可将其关于对象 j 的访问权 access(i,j)扩展到访问矩阵同一列中的其它域中，即为运行在其它域的进程也赋予关于同一对象的同样访问权限（access(k,j)）。

20. 如何利用拥有权来增、删某种访问权？

如果域 i 具有关于对象 j 的所有权，则运行在域 i 的进程可以增加或删除在 j 列的任何项中的任何访问权。换言之，该进程可以增加或删除在任何其它域中运行的进程关于对象 j 的任何访问权。

21. 增加控制权的主要目的是什么？试举例说明控制权的应用。

拷贝权和所有权均用于改变运行在不同域中的进程对同一对象的访问权，而控制权则用于改变某个域中运行进程关于不同对象的访问权。进一步说，若某域访问权 access(i,j)中含有控制权 C，则运行在 Di 域中的进程能改变运行在 Qj 域中的任何进程关于任何对象的任何访问权。

22. 什么是访问控制表？什么是访问权限表？

针对访问矩阵按列（对象）进行划分，为每一列建立一张访问控制表，同时将访问矩阵属于对应列的所有空项删除，故而访问控制表由有序对集{<域，权集>}所组成，可用于描述不同用户（进程）关于同一对象的不同访问权限集。针对访问矩阵按行（域）进行划分，为每一行建立一张访问权限表，其由有序对集{<对象，权集>}所组成。当域为用户(进程)，对象为文件时，访问权限表便可用来描述一个用户(进程)对每一个文件所能执行的一组操作。访问控制表也可用于定义各域关于某对象的缺省的访问权集，并作为资源能否使用的首要依据。

23. 系统如何利用访问控制表和访问权限表来实现对文件的保护？(P252)

答：我们可利用访问矩阵来描述系统的存取控制，访问矩阵的行代表域，列代表对象，矩阵中的每一项是由一组访问权组成。

访问控制表：是指对访问矩阵列（对象）进行划分，为每一列建立一张访问控制表 ACL。在该表中已经把矩阵总属于该列的所有空项删除，此时的访问控制表是由一有序对（域，权集）所组成。系统中配置了这种表后，当某用户（进程）要访问某资源时，通常，系统首先到缺省访问控制表中去查找该用户（进程）是否具有对指定资源进行访问的权力，如果找不到再到对应对象的访问控制表中去查找。

访问权限表：是指访问矩阵按行（即域）进行划分，由每一行构成一张访问权限表。这是一个由一个域对每一个对象可以执行的一组操作所构成的表。表中的每一项即为该域对某一对象的访问权限。如 UNIX 系统中：

类型	权力	对象
文件	RW—	指向文件的指针

首先访问权限表必须是安全的，系统对其使用了特殊的保护。

目前，大多数的系统都同时采用访问控制表和访问权限表，在系统中为每个对象配置一张访问控制表。当一个进程第一次试图访问一个对象时，必须先检查访问控制表，检查进程是否具有对该对象的访问权，如果无权访问该对象，由系统来拒绝该进程的访问，并构成一例外（异常）事件；否则（有权访问）便允许该进程访问该对象而为之建立一访问权限，并将它连接到该进程，以后进程便可直接利用这一返回的权限去访问该对象，这样便可快速地验证访问地合法性。当进程不再需要对该对象进行访问时，便可将访问权限取消。

24. 在对文件的四级安全管理中，每一级安全管理的主要用途是什么？(P253)

答：文件的四级安全管理措施：

（1） 系统级管理：主要任务时部允许未经核准的用户进入系统，从而也就防止了他人非法地使用系统中地各类资源。主要地方法有注册、登录、时限等等。

（2） 用户级安全管理：是为了给用户分配“文件访问权”而设计的。包括对所有用户进行分类、为指定用户分配文件访问权等。

（3） 目录级安全管理：是为了保护系统中的各种目录而设计的，它与用户权限无关。为保证目录的安全，规定只有系统核心才具有写目录的权利。通常，系统是分别为用户和目录独立地指定权限的。当一用户试图访问一目录时，核心将通过对用户访问权和目录中的访问权的比较后，用户才能获得有

效的访问权。

（4） 文件级安全管理：是通过系统管理员或文件主对文件属性的设置，来控制用户对文件的访问。用户对文件的访问，将由用户访问权、目录访问权限和文件属性三者的权限所确定。

第九章

1. 磁盘访问时间由哪几部分构成？每部分时间应如何估算？

磁盘访问时间包括以下三个部分：（1）寻道时间 Ts ，指把磁臂从当前位置移动到指定磁道上所经历的时间。该时间是启动磁盘的时间 s 与磁头移动 n 条磁道所花费的时间之和，即 Ts = m×n + s 。其中 m 是一常数，与磁盘驱动器的速度有关。（2）旋转延迟时间 Tr ，是指定扇区旋转到磁头下面所经历的时间。（3）传输时间 Tt ，指把数据从磁盘读出或向磁盘写入数据所经历的时间，其与每次所读/写的字节数 bytes 及旋转速度 r 有关，具体为 Tt = bytes / (r×bytesPerTrack)，其中 bytesPerTrack 为一条磁道上的字节数。当一次读/写的字节数相当于半条磁道上的字节数时，Tt 与 Tr 相同，也即 Tr = 1 / 2r。因此可将访问时间 Ta 表示为：Ta = Ts + 1/2r + bytes / (r×bytesPerTrack)。

2. 目前常用的磁盘调度算法有哪些？每种算法优先考虑的问题是什么？

目前常用的磁盘调度算法包括：（1）先来先服务调度算法 FCFS。根据进程请求访问磁盘的先后次序进行调度，其优点是公平、简单且每个进程的请求都能依次得到处理，不会出现某一进程的请求长期得不到满足的情况，但寻道时间可能较长。（2）最短寻道时间优先调度算法 SSTF。选择所要求访问磁道与磁头当前所在磁道距离最近的进程优先调度，但其并不能保证平均寻道时间最短。本算法具较好的寻道性能，但可能导致进程饥饿现象。（3）扫描算法 SCAN（又称为电梯调度算法），对最短寻道时间优先调度算法略加修改而形成。不仅考虑欲访问磁道与磁头当前所在磁道的间距，更优先考虑的是磁头当前移动的方向既能获得较好的寻道性，又能防止进程饥饿，广泛用于大、中、小型机及网络中。扫描算法存在的问题是：当磁头刚从里到外移动过某一磁道时，恰有一进程请求访问此磁道，该进程必须等待，待磁头从里向外，然后再从外向里扫描完所有要访问的磁道后，才处理该进程的请求，致使该进程的请求被严重推迟。（4）循环扫描算法 CSCAN。规定磁头单向移动，避免了扫描算法导致的某些进程磁盘请求的严重延迟。（5）N-步扫描算法。为克服前述 SSTF、SCAN、CSCAN 等调度算法都可能出现的磁臂停留在某处不动的情况即磁臂粘着现象，将磁盘请求队列分成若干个长度为 N 的子队列，按先来先服务算法依次处理这些子队列，而各队列分别以扫描算法进行处理。（6）FSCAN 算法，其实质为 N-步扫描算法的简化。具体而言，将磁盘请求队列分成两个子队列：①当前所有请求磁盘 I/O 的进程形成的队列，按扫描算法处理；②在扫描期间新出现的所有磁盘请求进程队列，本次扫描结束后②添加到①的队尾，从而使所有新要求都被推迟到下一次扫描时处理。

3. 磁盘空间连续分配的主要优缺点是什么？

磁盘空间连续分配要求为每一个文件分配一相邻接的盘块。在采用连续分配方式时，可把逻辑文件中的记录顺序地存储到邻接的各物理盘块中。这种分配方式保证了逻辑文件中的记录顺序与存储器中文件占用盘块的顺序一致性；但同时也会带来外部碎片问题，即伴随文件空间的分配和文件删除时的收回，将使磁盘空间不断分割而形成一系列较小的无法存储文件的连续区（当然可以通过紧凑方法加以处理，但系统开销很大）。归纳来说，连续分配的主要优点如下：（1）顺序访问容易。访问一个占有连续空间的文件，非常容易。同时连续分配也支持直接存取。（2）顺序访问速度快。因为由连续分配所装入的文件，其所占用的盘块可能是位于一条或几条相邻的磁道上，所以磁头的移动距离最少，访问速度快。其主要缺点包括：（1）要求有连续的存储空间，空间利用率低；（2）必须事先知道文件的长度。

4. 什么是隐式链接和显式链接？什么是文件分配表？

隐式链接和显式链接均为链接分配方式，支持离散分配，因而消除了外部碎片，外存空间利用率较高，能实现按需分配且无需事先知道文件长度，支持文件的动态增长，并方便了文件增、删、改。① 采用隐式链接分配方式时，通过每个盘块上的指针来实现同一文件多个离散盘块的链接，同时在文件目录的每个目录项中，都必须含有指向链接文件第一盘块和最后一个盘块的指针。隐式链接的主要问题是

只适合顺序访问，对随机存取极其低效；同时，由于其仅通过链接指针来实现离散各盘块的链接，所以只要其中任何一个指针出现问题，都会导致整条链的断开，因而可靠性较差。为了提高检索速度和减少指针所占用的存储空间，可将几个盘块组成一个簇，以簇为单位进行盘块分配。但又会带来内部碎片增大的缺点。② 显式链接是指用于链接文件各物理块的指针，显式地存放在内存的一张链接表中。该表是整个磁盘仅设置的一张表，以物理盘块号为表项序号，而以对应下一盘块号即链接指针作为表项内容。在该表中，凡是属于某一文件的第一个盘块号，或者说是每一个链的链首指针所对应的盘块号，均作为文件地址被填入相应文件的文件控制块 FCB 的“物理地址”字段中。由于查找记录的过程是在内存中进行的，因而不仅显著地提高了检索速度，而且大大减少了访问磁盘的次数。鉴于分配给文件的所有盘块号都放在该表中，故把该表称为文件分配表 FAT。

5. 假定盘块的大小为 1KB，对于 540MB 的硬盘，其文件分配表需占用多少存储空间？当硬盘容量为 1.2GB 时，文件分配表又需占用多少存储空间？

假定盘块的大小为 1KB。对于 540MB 的硬盘，共有盘块 540MB/1KB = 540K ∈(219, 220)，故文件分配表表项应取 20 位即 2.5B，所以其文件分配表需占用存储空间 540K×2.5B = 1350KB；当硬盘容量为 1.2GB 时，共有盘块 1.2GB /1KB = 1.2M ∈(220, 224) ，故文件分配表表项应取 24 位即 3B，所以其文件分配表需占用存储空间 1.2M×3B = 3.6MB。

6. 为什么要引入索引分配方式？其主要问题是什么？

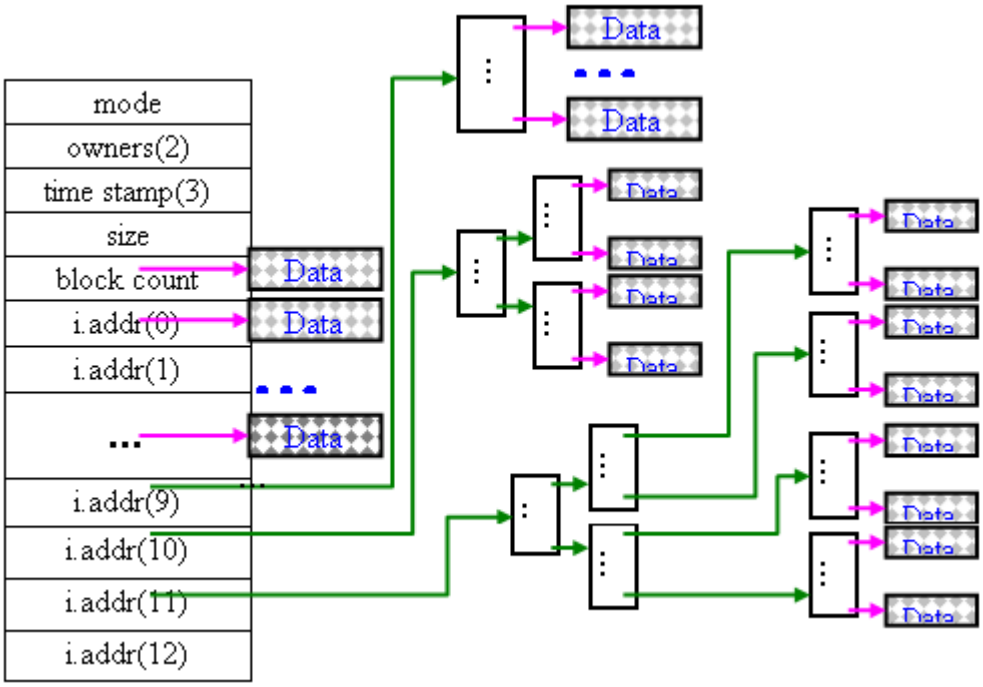
链接分配方式（特别是显式链接分配方式）虽然解决了连续分配方式存在的问题，但又出现了另外两个问题：（1）不能支持高效地直接存取，因为若对一个较大的文件进行直接存取，须首先在文件分配表中顺序地查找许多盘块号；（2）文件分配表需占用较大的内存空间。事实上，在打开某个文件时，只须把该文件占用的盘块编号调入内存即可，完全没有必要将整个 FAT 调入内存。为此，可将每个文件所对应的盘块号集中地存放一个所谓的索引块中，形成一张索引表，而在建立文件时应在其对应目录项中填上指向该索引块的指针。这便是所谓的索引分配方式。其存在的主要问题是可能要花费较多的外存空间，特别对于小文件来说，关于索引块的利用率是极低的。

7. 假如盘块的大小为 4KB，每个盘块号占 4 个字节，在两级索引分配时，允许的最大文件是多少？

假如盘块的大小为 4KB，每个盘块号占 4 个字节，则一个索引块可含 4KB/4B=1K 个盘块号，于是两级索引最多可含 1K×1K = 1M 个盘块号，因此，允许的最大文件长度为 4KB×1M = 4GB。

8. 详细说明 UNIX 系统采用的是何种磁盘分配方式？

UNIX 系统采用的是混合磁盘分配方式（参图示）。以 UNIX System V 为例，其索引结点中，共设置有 13 个地址项，具体分为以下三类地址：（1）直接地址。为提高对文件的检索速度，在索引结点中可设置 10 个直接地址项，即用 i.addr(0) ~ i.addr(9)来存放直接地址。换言之，各直接地址表项存放的是对应文件数据盘块的盘块号。设盘块大小为 4KB，盘块号占 4B，则支持长度在 4KB×10 = 40KB 以内的文件。（2）一次间接地址。对于大、中型文件，只采用直接地质是不现实的。因此，再利用索引结点的地址项 i.addr(10)来提供一次间接地址以指向对应文件的一级索引块，其实质是一级索引分配方式。鉴于一级索引块可含 4KB/4B = 1K 个盘块号，故支持长度在(4KB×1K=4MB)+40KB 以内的文件。（3）多次间接地址。当文件长度大于 4MB+40KB 时，系统还须采用二次甚至三次间址分配方式。具体而言，用 i.addr(11)、i.addr(12)分别指向对应文件的两级索引块和三级索引块，所以支持文件长度可达(4KB×1K×1K ×1K=4TB)+(4KB×1K×1K=4GB)+4MB+ 40KB。



9. 空闲磁盘空间的管理常采用哪几种方式？UNIX 系统采用的是何种方式？

空闲磁盘空间的管理常采用以下几种方法：（1）空闲表法，属于连续分配方式，它与内存管理中的动态分区分配方式相似。（2）空闲链表法，将所有空闲盘区链接成一条空闲链。根据构成链的基本元素不同，可分为空闲盘块链和空闲盘区链。（3）位示图法，利用二进制的一位来表示磁盘中每一个盘块的使用情况，磁盘上的所有盘块都有一个二进制位与之对应，从而由所有盘块所对应的位构成一个集合，即位示图。（4）成组链接法，结合空闲表法和空闲链表法而形成。UNIX 系统采用的是成组链接法。

10. 计算机系统利用下面的位示图来管理空闲盘块，盘块大小为 1KB，现要为某文件分配两个盘块，试说明盘块分配的具体过程。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

根据位示图为某文件分配两个盘块的具体过程如下：（1）顺序扫描位示图，找出两个其值均为空闲即0的二进制位 Map[3, 3]，Map[4, 7]；（2）将二进制位 Map[3, 3]和 Map[4, 7]的行/列号转换为与之对应的盘块号 35、55；（3）把盘块号为 35 和 55 的盘块分配给该文件，同时修改位示图中的二进制位 Map[3, 3]=1，Map[4, 7]=0。

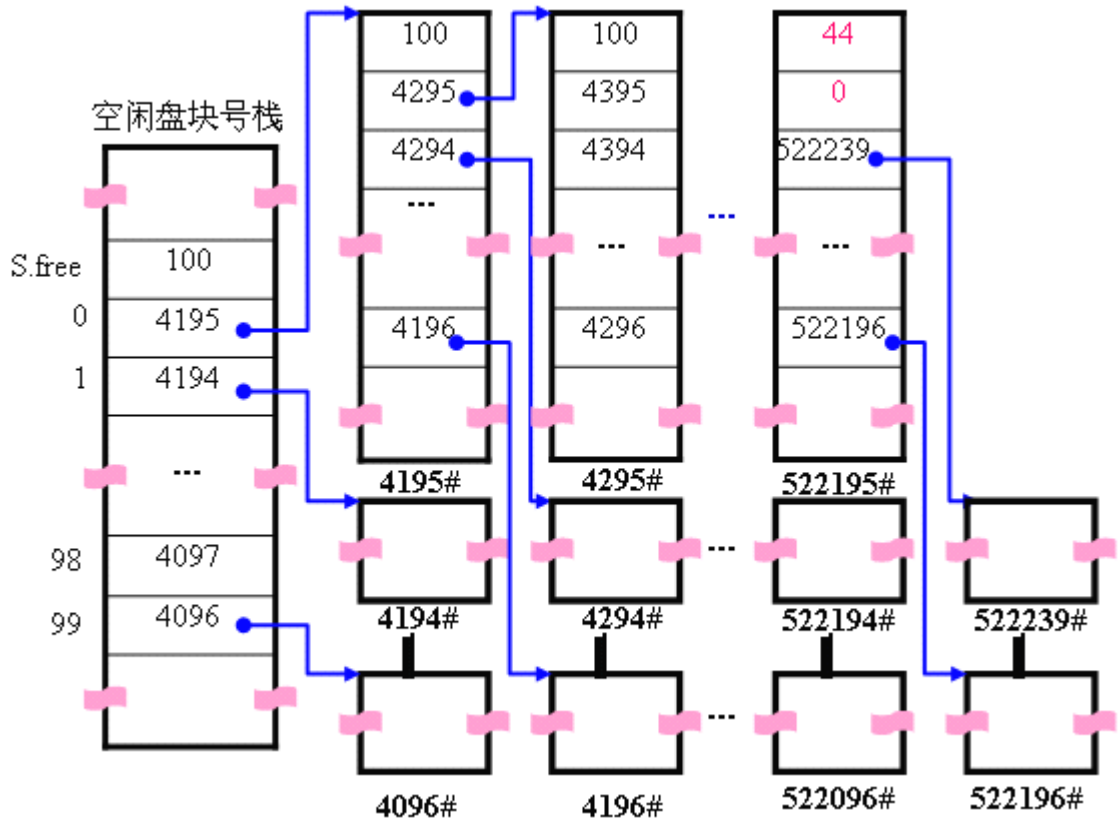
11. 在第一级磁盘容错技术中，包括那些容错措施？什么是写后读校验？

在第一级磁盘容错技术中，包括以下容错措施：（1）双份目录和双份文件分配表。在磁盘上存放的文件目录和文件分配表 FAT 均为文件管理所用的重要数据结构，所以为之建立备份。在系统每次加电启动时，都要对两份目录和两份 FAT 进行检查，以验证它们的一致性。（2）热修复重定向和写后读校验，二者均用于防止将数据写入有缺陷的盘块中。就热修复重定向而言，系统将一定的磁盘容量作为热修复重定向区，用于存放当发现盘块有缺陷时的待写数据，并对写入该区的所有数据进行登记，方便将来对数据进行访问。而写后读校验则是为了保证所有写入磁盘的数据都能写入到完好的盘块中，故在每次从内存缓冲区向磁盘中写入一个数据块后，应立即从磁盘上读出该数据块并送至另一缓冲区中，再将该缓冲区中内容与原内存缓冲区中在写后仍保留的数据进行比较，若两者一致，便认为此次写入成功，可继续写入下一个盘块；否则，则重写。若重写后两者仍不一致，则认为该盘块有缺陷，此时

便将应写入该盘块的数据写入热修复重定向区中，并将该损坏盘块的地址，记录在坏盘块表中。

12．某计算机系统磁盘容量为 520MB，盘块大小为 1KB。其中前 4MB 用于存放索引结点等，后 10MB 用作对换区，采用成组链接法管理外存空间，每组 100 个盘块。试画出外存尚未使用的成组链接图。

根据题意，该计算机系统尚未使用的外存空间为（520MB－4MB－10MB = 506MB），即 506K（也就是 518144）个盘块，其盘块号为 4K#~510K#（即 4096# ~ 522239#）。而每组 100 个盘块，故共有 5182 个盘块组，其中最后一个盘块组含 44 个盘块。因此，外存尚未使用的成组链接图如下所示：



13．在第二级磁盘容错技术中，包括那些容错措施？

在第二级磁盘容错技术中，包括以下容错措施：（1）磁盘镜像。在同一磁盘控制器下增设一个完全相同的磁盘驱动器，在每次向文件服务器的主磁盘写入数据后，都要采用写后读校验方式，将数据再同样地写到备份磁盘上，使二者具有完全相同的位像图。（2）磁盘双工。将两台磁盘驱动器分别接到两个磁盘控制器上，同样使这两台磁盘机镜像成对，从而在磁盘控制器发生故障时，起到数据保护作用。在磁盘双工时，由于每一个磁盘都有着自己的独立通道，故可以同时（并行）地将数据写入磁盘。在读入数据时，可采用分离搜索技术，从响应快的通道上取得数据，因而加快了对数据的读取速度。

14．廉价磁盘冗余阵列是如何提高对磁盘的访问速度和可靠性的？

廉价磁盘冗余阵列 RAID 通过利用一台磁盘阵列控制器来统一管理和控制一组磁盘驱动器，从而组成一个高度可靠的、快速的大容量磁盘系统。为了提高对磁盘的访问速度，其采用了并行交叉存取技术。具体而言，在该系统中，有多台磁盘驱动器，系统将每一盘块中的数据分为若干个盘块数据，并把每一个子盘块的数据分别存储到各个不同的磁盘中的相同位置。当要将一个盘块中的数据传送到内存时，采取并行传输的方式，将各个盘块中的子盘块数据同时向内存中传输，从而使传输时间大大减少。进一步说，RAID 分为 0~7 级，这里主要介绍以下六级：（1）RAID 0 级仅提供并行交叉存取，虽能有效提高磁盘的 I/O 速度，但无冗余校验功能，致使磁盘系统的可靠性不好。（2）RAID 1 级具有磁盘镜像功能，可利用并行读、写特性，将数据分块并行同时写入主盘和镜像盘，故比传统的镜像盘速度快，但它的磁盘容量的利用率只有 50%。（3）RAID 3 级具有并行传输功能，它利用一台奇偶校验盘来完成容错功能，同磁盘镜像相比，它减少了所需要的冗余磁盘数，常用于科学计算和图像处理。（4）RAID 5 级具有独立传送功能，每个驱动器都有各自独立的数据通道，独立地进行读写且无专门的校验盘，用

来进行纠错的校验信息，是以螺旋方式散布在所有数据盘上，常用于 I/O 较频繁的事务处理。(5) RAID 6 级的阵列中，设置了一个专用的、可快速访问的异步校验盘，该盘具有独立的数据访问通路，具有比 RAID 3 和 RAID 5 更好的性能。（6）RAID 7 级是对 RAID6 级的改进，在该阵列中的所有磁盘都具有较高的传输速度、有着优异的性能，是目前最高档次的磁盘阵列。

15. 常用的后备系统有哪些类型？其中最常用的是哪一类？

常用的后备系统有三类：磁带机、磁盘机和光盘机。其中，过去及现在用的最多和最常用的是磁带机。

16. 可采用哪些方法将磁盘上的数据拷贝到后备设备上？

为将磁盘上的数据拷贝到后备设备上，可采用两种方法：（1）完全转储法，指定期地将磁盘上的整个文件系统，拷贝到后备系统上。此方法虽简单，但效率低。（2）增量转储法。为了实现增量转储，在系统中应配置一张转储时间表，在其中记录下每个文件最近一次的转储时间。

17. 设计磁盘高速缓存的置换算法时，应考虑哪些问题？

设计磁盘高速缓存的置换算法时，除了考虑到最近最久未使用这一原则外，还应考虑以下几点：（1）访问速率；（2）可预见性；（3）数据一致性。基于上述考虑，某些系统中便将高速缓存中的所有盘块数据拉成一条 LRU 链，对于那些会严重影响数据一致性的盘块数据和很久都可能不再使用的盘块数据，都放在 LRU 链的头部，使它们能够被优先写回磁盘，以减少数据发生不一致性的概率或可以尽量腾出高速缓存的空间；对于那些可能在不久以后便要再次使用的盘块数据，应挂在 LRU 链的尾部，以便在不久以后需要时，只要该数据块尚未从链中移至链首而被写回磁盘，便可直接到高速缓存即 LRU 链中区找到它们和进行访问。

18. 改善文件系统的性能可通过哪些途径？

改善文件系统的性能主要通过以下途径来实现：（1）关于文件访问的快速性，应改进文件目录结构及目录检索方法，选择适当的文件结构，提高磁盘的 I/O 速度；（2）关于数据的可共享性，可基于索引结点、基本目录、符号链等来展开；（3）关于文件系统使用的方便性，应从按名存取、追求接口简单且功能强大等方面着手解决；（4）关于数据的安全性，应从健全多级安全管理体系及存取控制机制、加强系统容错技术等入手来保证；（5）关于数据的一致性，应基于事务、检查点、并发控制以及盘块、联接计数和重复文件的一致性控制检查机制来加强。

19. 什么是虚拟盘？它具有哪些优缺点？

虚拟盘是指利用内存空间去仿真磁盘，又称为 RAM 盘。该盘的设备驱动程序可以接受所有标准的磁盘操作，但这些操作的执行，不是在磁盘上而是在内存中，所以会略微快些，且相关过程对用户透明。虚拟盘存在的主要问题是，由于内存为易失性存储器，故一旦系统或电源发生故障或系统再启动时，原来保存在虚拟盘中的数据将会丢失。因此，虚拟盘通常用于存放临时文件，如编译程序所产生的目标程序等。虚拟盘与磁盘高速缓存的主要区别在于，虚拟盘中的内容完全由用户控制，而高速磁盘缓存中的内容则是由操作系统控制的。

20. 什么是事务？如何保证事务的原子性？

事务是用于访问和修改各种数据项（可分散在多个文件中）的一个程序单位，其可被看作是一系列的读写操作，且具有“原子性”的特性。为保证事务的原子性，只有当所有操作全部完成时，才进行提交操作 Commit Operation 来终止事务；而只要有一个读或写操作失败，也即一个夭折的事务，应执行夭折操作，进一步说，由于该事务通常已经执行了一些操作，所以已对某些数据项进行了修改，为使夭折的事务不会引起数据的不一致性，须将该事务内刚被修改过的数据项恢复成原来的情况，使系统中数据项与该事务未执行时的数据项完全相同。为了实现上述的原子修改，通常需借助于称为事务记录（或运行记录）的数据机构和有关的恢复算法来实现。事务记录用于记录事务运行时数据项修改的全部信息，支持原子修改，存放于稳定存储器中。

21. 引入检查点的目的是什么？引入检查点后是如何进行恢复处理的？

引入检查点的主要目的是使对事务记录表中事务记录的清理工作经常化，即每隔一定时间，便扫描事务记录表，同时将驻留在易失性存储器即内存中当前事务记录表的全部记录和<检查点>记录及所有已修改数据输出到稳定存储器中；并且每当出现一个<检查点>记录时，应利用 Redo 和 Undo 过程执行恢复操作。引入检查点后的恢复例程首先查找事务记录表，确定在最近检查点以前开始执行的最后事务 Ti；在找到这样的事务后，再返回到事务记录表，便可找到第一个检查点记录，进而从该检查点开始

依次搜索各个事务记录，并基于 Redo 和 Undo 过程执行恢复操作。具体而言，针对 Ti 以后开始执行的事务集 T 中的事务 Tk，区别不同情况分别执行恢复操作 Redo(Tk) /Undo(Tk)：（1）如果在事务记录表中出现了<Tk 提交>记录，则执行 Redo（Tk）操作；（2）如果在事务记录表中未出现<Tk 提交>记录，则执行 Undo（Tk）操作。

22. 为什么引入共享锁？如何用互斥锁或共享锁来实现事务的顺序性？

在多用户系统中，可能有多个用户在执行事务。由于事务的原子性，各个事务的执行必然是按某种次序依次执行的，只有在一个事务执行完后，才允许另外一个事务执行，即事务对数据项的修改是互斥的。这也即事务的顺序性。实现事务顺序性的一种最简单的方法是为每个共享对象（文件、记录或数据项）设置一个互斥锁，同时要求事务对对象的访问以首先获得该对象的互斥锁为前提条件。进一步说，当一个事务 Ti 获得某共享对象的互斥锁后，便可对该对象执行读写操作；而其它事务由于不能获得互斥锁故不能访问该对象。特别地，如果 Ti 需要对一批对象进行访问，为了保证事务操作的原子性，Ti 应首先获得这一批对象的互斥锁，以将它们全部锁住。如果成功，便可对这一批对象执行读写操作，操作完毕后释放所有互斥锁；而如果不成功，即这一批对象中至少存在某个对象已被其它事务锁住，则应将那些刚被 Ti 锁住的对象进行开锁，宣布此次事务运行失败，而不致引发数据变化。必须指出，基于互斥锁实现顺序性的方法尽管简单易行，但却存在效率不高的问题。因为一个共享文件虽然只允许一个事务去写，但却允许多个事务同时读，而利用互斥锁来锁住文件时，则不加区分地只允许一个事务进行读或写操作。为了提高运行效率，应引入另外一种形式的锁即共享锁。具体而言，如果事务Ti 要对 Q 执行读操作，则只需获得 Q 的共享锁即可；而如果事务 Ti 获得了某对象 Q 的共享锁，仅仅允许 Ti 去对对象 Q 执行读操作，而不允许写操作。当事务 Ti 试图去获取对象 Q 的共享锁时，Q 已被互斥锁锁住，则 Ti 必须等待；否则，便可获得共享锁而对 Q 执行读操作。但如果 Ti 要对 Q 执行写操作，则它还须去获得互斥锁。若失败，则等待；否则，可获得互斥锁而对 Q 执行写操作。

23. 当系统中有重复文件时，应如何保证其一致性？

当系统中有重复文件时，可采用两种方法来保证和实现文件数据的一致性：（1）当一个文件被修改后，可查找文件目录，以得到其另外几个拷贝的索引结点号，从这些索引结点中找到各个拷贝的物理位置，然后对它们进行同样的修改；（2）为新修改的文件建立几个拷贝，并用它去替换原有文件拷贝。

24. 如何检查盘块号的一致性？检查时可能出现哪几种情况？

盘块是用于存储文件的物理空间，用于描述盘块的使用情况的数据结构包括两类：(1)空闲盘块表（链）记录了所有尚未分配和使用的空闲盘块编号；（2）文件分配表则用来记录已分配盘块的使用情况。为了保证盘块描述用数据结构的一致性，可在每次启动系统时通过用软件方法构建基于盘块号的一个计数器表来检查空闲盘块表（链）和文件分配表之间的一致性。在计数器表中，每个盘块号占一个表项，可有 0，1，---，（N-1）项，N 为盘块总数。每一表项中包含两个计数器，分别用作空闲盘块号计数器和数据盘块号计数器。在对盘块的数据结构进行检查时，应该先将计数器表中的所有表项初始化为 0；然后，用 N 个空闲盘块号计数器组成的第一组计数器，来对从空闲盘块表（链）中读出的盘块号进行计数；再用 N 个数据盘块号计数器组成的第二组计数器，去对从文件分配表中读出的、已分配给文件使用的盘块号进行计数。如果情况是正常的，则上述两组计数器中对应每个盘块号的空闲盘块号计数值和数据盘块号计数值应当互补，也就是说，当某个盘块号在第一组计数器中进行了计数使该盘块号计数器计为 1，则在第二组计数器中相应盘块号的计数器内容必为 0；反之亦然。但如果情况并非如此，则说明发生了错误：（1）两组计数器关于对应某盘块号 K 的计数值均为 0，说明该盘块未被利用，相应解决方案为在空闲盘块表（链）中增加一个盘块号 K；（2）空闲盘块号计数器关于某盘块号 K 的计数值为 2，而数据盘块号计数器关于盘块号 K 的计数值为 0，说明盘块号 K 在空闲盘块表（链）中出现了两次，应从中删除一个多余的空闲盘块号 K；（3）空闲盘块号计数器关于某盘块号 K 的计数值为0，而数据盘块号计数器关于盘块号 K 的计数值不小于 2，此种错误比较严重，必须立即报告系统和加以处理。

第十章

1. 操作系统包括哪几种类型的用户接口？它们分别提供给谁使用？

操作系统包括三种类型的用户接口：命令接口（具体又可分为联机命令接口与脱机命令接口）、程序接口及图形化用户接口。其中，命令接口和图形化用户接口支持用户直接通过终端来使用计算机系统，而程序接口则提供给用户在编制程序时使用。

2. 联机命令接口由哪些部分构成？

联机命令接口由一组联机命令、终端处理程序和命令解释程序构成。

3. 联机命令通常包含哪些类型？每种类型又包含哪些主要命令？

联机命令通常包含哪些类型？每种类型又包含哪些主要命令？联机命令通常包含如下类型：（1）系统访问类，主要是注册命令；（2）磁盘操作类，包括磁盘格式化、软盘复制、软盘比较及备份等命令；（3）文件操作类，包括文件内容显示、文件拷贝、文件比较、文件重命名、文件删除等命令；（4）目录操作类，包括子目录建立、目录显示、子目录删除、目录结构显示、当前目录改变等命令；（5）通信类命令；（6）其它命令，包括输入输出重定向、管道联接、过滤命令及批处理命令等。

4. 什么是输入输出重定向？试举例说明。

通常，命令的输入取自标准输入设备即键盘，而命令的输出则送往标准输出设备即显示终端。如果在命令中设置输出定向“〉”，其后接文件名或设备名，则表示命令的输出改向，并送到指定文件或设备上；类似地，在命令中设置输入重定向“〈”，则不再是从键盘而是从重定向符左边的参数指定的文件或设备上取得输入信息。这便是输入输出的重定向。

5. 什么是管道联接？试举例说明。

管道联接是指把第一个命令的输出作为第二个命令的输入，类似地又把第二个命令的输出作为第三条命令的输入，以此类推，这样由两条以上的命令可形成一条管道。在 MS-DOS 和 UNIX 中，都用"|"作为管道符号。其一般格式为：command1 |command2 | … |commandn

6. 终端处理程序的主要作用是什么？它应具备哪些功能？

终端处理程序主要用于实现人机交互，它应具备以下功能：（1）接收用户从终端上键入的字符；（2）管理字符缓冲，以暂存所接收的字符；（3）将用户键入字符回送屏幕显示；（4）提供屏幕编辑（编辑键）；（5）特殊字符处理（中断/停止或恢复上卷）。

7. 命令解释程序的主要作用是什么？

命令解释程序的主要作用是：在屏幕上产生提示符，请用户输入命令，然后读入命令、识别命令，并转至相应的命令处理程序入口地址，把控制权交给该处理程序去执行，最后将有关处理结果（包括出错信息）送屏幕显示。

8. 试说明 MS-DOS 的命令解释程序 COMMAND.COM 的工作流程。

MS-DOS 命令解释程序 COMMAND.COM 的主要工作流程如下：（1）系统接通电源或复位，初始化部分获得控制权，对整个系统完成初始化工作，并自动执行 Autoexec.bat 文件，然后把控制权交给暂存部分，后者给出提示符并等待和接收用户键入命令；（2）暂存部分读入键盘缓冲区中的命令，判别其文件名、扩展名及驱动器名是否正确，若有错则给定出错信息后返回；无错的情况下才查找和识别该命令；（3）若该命令为内部命令，暂存部分定会在命令表格中找到该命令，便可从对应表项中获得该命令处理程序的入口地址，并把控制权交给该程序去执行；若键入命令为外部指令，则暂存部分应为之建立命令行，通过执行系统调用 exec 装入其命令处理程序，并得到对应基地址，把控制权交由该程序执行；若键入命令非法，则出错返回；（4）命令完成后，控制权重新交给暂存部分给出提示符并等待和接收用户键入命令，然后转（2）。

9. 试比较一般的过程调用和系统调用。

系统调用本质上是一种过程调用，但它是一种特殊的过程调用，与一般的过程调用存在以下几方面的差别：（1）运行在不同的系统状态。一般的过程调用，其调用过程和被调用过程或者均为用户程序，或者均为系统程序，所以都运行在同一系统执行状态（用户态或系统态）下；而系统调用的调用过程是运行在用户态下的用户程序，被调用过程是运行在系统态下的系统程序。（2）软中断进入机制。一般的过程调用可直接由调用过程转向被调用过程；而执行系统调用时，由于调用过程和被调用过程是处于不同的系统状态，所以不允许由调用过程直接转向被调用过程，而通常是通过软中断机制，先进入操作系统内核，经内核分析后，才能转向相应的命令处理程序。（3）返回及重新调度问题。对于一

般的过程调用，在被调用过程执行完后，将返回到调用过程继续执行；对于系统调用则不是这样，在被调用过程执行完后，要对系统中所有要求运行的进程进行重新调度。特别地，在采用了抢占式剥夺调度的系统中，重新调度将基于优先权分析来进行，于是只有当调用进程仍具有最高优先权时，才会返回到调用过程继续执行；否则其将会被放入就绪队列，而执行权利交由具最高优先权的过程优先执行。（4）嵌套调用。与一般过程类似，系统调用也允许嵌套调用，即在一个被调用过程执行期间，还可以再利用系统调用命令去调用另一个系统过程，注意是系统过程而不是用户过程。

10. 什么是系统调用？它都有哪些类型？

通常，在操作系统内核设置有一组用于实现各种系统功能的子程序（过程），并将它们提供给用户程序调用。每当用户在程序中需要操作系统提供某种服务时，便可利用一条系统调用命令，去调用所需的系统过程。这即所谓的系统调用。系统调用的主要类型包括：（1）进程控制类，主要用于进程的创建和终止、对子进程结束的等待、进程映像的替换、进程数据段大小的改变以及关于进程标识符或指定进程属性的获得等；（2）文件操纵类，主要用于文件的创建、打开、关闭、读/写及文件读写指针的移动和文件属性的修改，目录的创建及关于目录、特别文件或普通文件的索引结点的建立等；（3）进程通信类，用于实现各种类型的通信机制如消息传递、共享存储区及信息量集机制等；（4）信息维护类，用于实现关于日期和时间及其它系统相关信息的设置和获得。

11. 如何设置系统调用所需的参数？

设置系统调用所需参数包括两种方式：（1）直接将参数送入相应的寄存器中，这是一种最简单的方式。其主要问题是由于寄存器数量有限，从而限制了设置参数的数目。（2）参数表方式，也即将系统调用所需的参数，放入一张参数表中，再将指向该参数表的指针放在某个规定的寄存器中。本方式具体又可分为直接方式和间接方式两类。

12. 试说明系统调用的一般处理过程。

系统调用的一般处理过程分为三步：（1）设置系统调用号和参数。（2）对系统调用命令进行一般性处理，如保护 CPU 现场，将处理机状态字 PSW、程序计数器 PC、系统调用号、用户栈指针以及通用寄存器等压入堆栈，将用户定义参数传送至指定位置保存起来等。不同系统具体处理方式往往不同，在 UNIX 系统中是执行 CHMK 命令，并将参数表中的参数传到 User 结构的 U.U-arg()中；而在 MS-DOS 中则是执行 INT21 软中断。（3）根据系统调用入口表及具体的系统调用命令转至对应命令处理程序执行具体处理。

第十一章

第十二章

第十三章

1. UNIX 系统有哪些基本特征？

a. 开放性；
b. 多用户，多任务环境；
c. 功能强大，实现高效；
d. 提供了丰富的网络功能.

2. UNIX 系统核心分成哪两大部分？各包含哪些功能？

a. UNIX 系统核心分为进程控制子系统部分和文件子系统部分；
b. 进程控制子系统包含进程控制，进程通信，存贮器管理和进程调度功能；文件子系统包含文件管理，高速缓冲机制和设备驱动程序的功能.

3. UNIX 系统中的 PCB 含哪几部分？并用图来说明它们之间的关系.

a. UNIX 系统中的 PCB 含四部分:进程表项，U 区，进程区表和系统区表项；
b. 图见 P396.

4. 进程映象含哪几部分？其中系统级上下文的动态部分的作用是什么？

a. 进程映象(Process Image)包含三部分：用户级上下文，寄存器上下文和系统级上下文；

b. 系统级上下文的动态部分包含核心栈和若干层寄存器上下文，它的作用是当因中断或系统调用而进入核心态时，核心把一个寄存器上下文压入核心栈，退出系统调用时，核心又将弹出一个寄存器上下文，在进行上下文切换时，核心将压入老进程的上下文层，而弹出新进程的上下文层.

5. 在 UNIX 系统中，用于进程控制的系统调用有哪些(主要的)？它们的主要功能是什么？

a. fork，用于创建一个新进程；

b. exec，改变进程的原有代码；

c. exit，实现进程的自我终止；

d. wait，将调用进程挂起，等待子进程终止；

e. getpid，获取进程标志符；

f. nice，改变进程的优先级.

6. 为创建一个新进程，需做哪些工作？

a. 为新进程分配一进程表项和进程标志符；

b. 检查同时运行的进程数目；

c. 拷贝进程表项中的数据；

d. 子进程继承父进程的所有文件；

e. 为子进程创建进程上下文；

f. 子进程执行.

7. 为何要采取进程自我终止方式？如何实现 exit？

a. 为了及时回收进程所占用的资源，并减少父进程的干预，UNIX 系统利用 exit 来实现进程的自我终止；

b. 实现 exit，核心应该做的工作是：

关闭软中断；

回收资源；

写记帐信息；

置进程为"僵死状态"。

8. UNIX 系统采用什么样的进程调度算法？其优先级是如何计算的？

a. UNIX 系统采用的是多级反馈队列轮转调度算法；

b. 每隔 1 秒，核心按如下公式重新计算用户优先数：优先数=(最近使用 CPU 的时间/2)+基本用户优先数.

9. 试说明信号与中断两种机制间的异同处？

a. 相似处：

信号和中断都采用了相同的异步通信方式；

当检测出有信号或中断请求时，都是暂停正在执行的程序而转去执行相应的处理程序；

两者都是在处理完毕后返回到原来的断点；

对信号或中断都可进行屏蔽；

b. 差异处：

中断有优先级，而信号没有优先级，即所有信号都是平等的；

信号处理程序是在用户态下运行的，而中断处理程序则是在核心态下运行的；

中断响应是及时的，而信号响应通常都有较大的时间延迟.

10 扼要说明信号机制中信号的发送和对信号的处理功能？

a. 信号的发送是指由发送进程把信号送到指定进程的信号域的某一位上；

b. 对于对信号的处理功能：

首先，

利用系统调用 signal(sig,func)预置对信号的处理方式，func=1 时，该类信号被屏蔽；

func=0 时，进程收到信号后终止自己；

func 为非 0，非 1 类整数时，func 的值即作为信号处理程序的指针.

然后，

如果进程收到的软中断是一个已决定要忽略的信号(func=1)，进程不作任何处理返回；

进程收到软中断后便退出(func=0)；

执行用于设置的软中断处理程序.

11 什么是管道？无名管道和有名管道的主要差别是什么？

a. 管道是指能够连接一个写进程和一个读进程的，并允许它们以生产者-消费者方式进行通信的一个共享文件，又称为 pipe 文件；

b. 无名管道是一个临时文件，是利用系统调用 pipe()建立起来的无名文件，没有路径名，只有调用 pipe 的进程及其子孙进程才能识别此文件描述符而利用该文件(管道)进行通信；有名管道是利用 mknod 系统调用建立的，是可以在文件系统中长期存在的，既有路径名的文件，其它进程可以知道其存在，并利用该路径名来访问该文件.

12 读，写管道时应遵循哪些规则？

a. 对 pipe 文件大小的限制；

b. 进程互斥；

c. 进程写管道时，检查是否有足够的空间存放要写的数据，若有，则写入，若无，则由核心对该索引结点做出标志，然后让写进程睡眠等待，直到读进程读走数据后，再将写等待进程唤醒；

d. 进程读管道时，检查是否有足够的要读的数据，若有，则进程从读指针的初始值处去读数据，每读出一块后，便增加地址项的大小，读结束后由核心修改索引结点中的读指针，并唤醒所有等待的写进程,若无，则在读完后，进程暂时进入睡眠等待，直到写进程又将数据写入管道后，再将读进程唤醒.

13 在消息机制中，有哪些系统调用？并说明它们的用途.

在 UNIX 中，消息机制向用户提供了四个系统调用：

a. msgget()，用来建立一消息队列，或者获取一消息队列的描述符；

b. msgsnd()，用于向指定的消息队列发送一个消息，并将该消息链接到该消息队列的尾部；

c. msgrcv()，用于从指定的消息队列中接收指定类型的消息；

d. msgctl()，用来读取消息队列的状态信息并进行修改.

14 在共享存储区机制中，有哪些系统调用？并扼要说明它们的用途.

a. shmget()，建立一共享存储区；

b. shmat()，将共享存储区附接到进程的虚地址空间上；

c. shmdt()，把共享存储区与新进程断开；

d. shmct()，对共享存储区的状态信息进行读取和修改，也可以断开进程与共享存储区的连接.

15 核心在执行 shmget 系统调用时，需完成哪些工作？

a. 首先检查共享存储区表，若找到指定 key 的表项，表明该共享区已经建立，此时返回该表项的描述符 shmid；

b. 若未找到指定的 key 表项，而 flag 标志又为 IPC_CREAT，且参数 size 值在系统限制值内，则分配一系统空闲区作为共享区的页表区，分配响应的内存块，再将这些块号填入页表中；

c. 核心在共享存储区和系统区表中，为新建立的共享区分配一空表项，并在共享存储区表填上存储区

的关键字及其大小，共享区页表的始址，指向系统区表项的指针等，最后返回共享存储区的描述符---shmid.

16 在信号量集机制中，有哪些系统调用？并说明它们的用途.

a. semget()，建立信号量集；

b. semop()，对信号量进行操作.

17 核心是如何对信号量进行操纵的？

a. 核心根据 sem_op 来改变信号量的值，可分为 3 种情况；

b. sem_op 的值为正，则将其值加到信号量的值上，它相当于通常的 V 操作；

c. sem_op 的值为负，相当于 P 操作，若信号量的值大于操作值的绝对值，则核心将一个负整数加到信号量值上，否则，核心将已经操作了的信号量，恢复到系统调用开始时的值；

d. 若(sem_flg&IPC_NOWAIT)为真，便立即返回，否则，让进程睡眠等待.

18 为实现请求调页管理，在 UNIX 系统中，配置了哪些数据结构？

a. 页表；

b. 磁盘块描述表；

c. 页框数据表；

d. 对换使用表.

19 在 UNIX 系统中，如何改变有效页的年龄？并用实例说明之.

a. 一个页可计数的最大年龄，取决于它的硬件设施；

b. 对于只设置两位作为年龄域时，其有效页的年龄只能取值为 0，1，2，3，当该页的年龄为 0，1，2 时，该页处于不可换出状态，而当其年龄达到 3 时，则可为换出状态，每当内存中的空闲页面数低于某规定的低限时，核心便唤醒换页进程，又换页进程取检查内存中的每一个活动的，非上锁的区，对所有有效区的年龄字段加 1，对于那些年龄已增至 3 的页便不再加 1，而是将它们换出，如果这种页已被进程访问过，便将年龄域中的年龄降为 0.

20 当需访问的缺页是在可执行文件上或在对换设备上时，应如何将它调入内存？

核心先为缺页分配一内存页，修改该页表项，使之指向内存页，并将页面数据表项放入相应的散列队列

中，然后把该页从对换设备上调入内存，当 I/O 操作完成时，核心把请求调入该页的进程唤醒.

21 在将一页换出时，可分为哪几种情况？应如何处理这些情况？

a. 若在对换设备上已有被换出页的拷贝，且被换出页的内容未被修改，则此时核心不必将该页重写回对换设备上，而只需将该页的页表项中的有效位清零，并将页框数据表项中的引用计数减 1，最后将该页表项放入空闲页链表中；

b. 若在对换设备上没有被换出的拷贝，则换出进程应将该页写到对换设备上，可采用页面链集中写入；

c. 在对换设备上已有换出页的副本，但该页内容已被修改过，此时核心将该页在对换设备上的原有空间释放，再重新将该页拷贝到对换设备上，使在对换设备上的拷贝内容总是最新的.

第十四章

第十五章

第十六章