第一章

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．在生产者一消费者问题中，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，引入分页存储管理(估计印错了，是分段存储管理)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

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．