1.操作系统的作用

答：操作系统提供了程序执行的环境。它的职能是管理和控制计算机系统中的所有软硬件资源，合理的组织计算机工作流程，并为用户提供一个良好的工作环境与友好的接口。

2.操作系统包括哪些功能

答：

存储器管理功能，主要包括：内存分配、地址映射、内存保护和内存扩充。

处理机管理功能，其功能包括：作业和进程调度，进程控制和进程通信。

设备管理功能，主要包括：缓冲区管理、设备分配、设备驱动和设备无关性（设备处理）。

文件管理功能，其功能包括：文件存储空间的管理、文件操作的一般管理、目录管理、文件的读写管理，存取控制和保护。

用户接口：命令接口、程序接口、图形接口

3.核心模式和用户模式

答：核心模式一般指操作系统管理程序运行的状态，具有较高的特权级别。

用户模式一般指用户程序运行时的状态，具有较低的特权级别。

当处理器处于管态时全部指令（包括特权指令）可以执行，可使用所有资源，并具有改变处理器状态的能力。当处理器处于用户模式时，就只能执行非特权指令。特权级别不同，可运行指令集合也不同。特权级别越高，可以运行指令集合越大。高特权级别对应的可运行指令集合包含低特权级的。核心模式到用户模式的唯一途径是通过中断。

4.操作系统提供的服务有哪些

答：程序执行、I/O 操作、文件系统处理、通信、错误检测、资源分配、户管理、保护

5.系统调用的工作机制

用户在需要执行特权指令时，调用系统调用，陷入内核（不同的任务，所对应调用的系统调用号也不同，在调用系统调用陷入内核时，会同时向OS内核传入一个系统调用号i）

进入内核后，根据i查找系统调用表，找到调用号为i的系统调用的处理代码

内核执行完系统调用处理代码后，从核心态返回用户态

6操作系统的结构有哪些，各自优缺点

答：1.简单结构 2. 层次化设计3.微内核

要求：能用简单的语言说明不同结构操作系统的特点

7虚拟机的优点

答：虚拟机技术主要有两个优点。

首先，通过完全的保护系统资源，虚拟机提供了一个健壮的安全保护层。

其次，虚拟机允许在不干扰正常的系统操作的情况下进行系统开发。

1进程的概念及组成。

概念：进程是指在系统中能独立运行并作为资源分配的基本单位，它是由一组机器指令、数据和堆栈等组成的，是一个能独立运行的活动实体。多个进程间可以并发执行和交换信息。一个进程在运行时需要一定的资源，如CPU、存储空间及I/O设备等。

组成：

（1）进程标识符：它是惟一的标志对应进程的一个标志符或数字；

（2）处理机状态：包括是处理机的各种寄存器内容信息；

（3）进程调度信息：表明该进程的执行状态；调度优先权：表示进程获取CPU的优先级别；进程之间通信信息：反映该进程与哪些进程有什么样的通信关系；

（4）进程控制信息：被保护的信息有：程序计数器程序状态字，各工作寄存器的内容等；资源需求、分配和控制方面的信息；进程实体信息：指出该进程的程序和数据的存储情况，在内存或外存的地址、大小等；族系关系：反映父子进程的隶属关系；其它信息：如文件信息、工作单位等。

进程的基本状态及转换条件

状态：

创建：进程正被创建。

运行： （进程的）指令正被执行。

等待：进程正在等待发生一些事件（如I/O 完成或接收一个信号）。

就绪：进程正等待分配处理器。

终止：进程结束运行

转换：

(1) 就绪→运行：进程具备运行条件，当进程调度程序选择了进程后，便将其转入运行状态。

(2) 运行→阻塞：进程需要等待某种事件的发生，如执行了输入输出指令，或者请求资源得不到满足时，进程转阻塞状态。

(3) 阻塞→就绪：进程等待的I/O已完成，或者请求的资源得到满足，进程转为就绪状态。

(4) 创建→就绪：进程尚不具备运行条件，所需的资源尚未得到满足。当进程创建完成后，进程可转入就绪状态。

(5) 运行→延迟：进程运行过程中，因某种原因需要延迟运算，则设定好延迟时间后被转入延迟状态。

(6) 运行→完成：进程运行时遇到结束指令后，被转入完成状态。

3进程控制块（PCB）的作用

答：进程控制块是进程组成中最关键的部分。每个进程有惟一的进程控制块。操作系统根据PCB对进程实施控制和管理。进程的动态、并发等特征是利用PCB表现出来的。PCB是进程存在的惟一标志。

4进程调度的类型（长，中，短）及调度的过程（上下文切换）

（1）高级调度（high Level Scheduling）：又称为作业调度或者长程调度（longTerm Scheduling）,其主要功能是根据某种算法，把外存上处于后备队列中的那些作业调入内存，也就是说，它调度对象是作业。P84

（2）低级调度（low Level Scheduling）称为进程调度或短程调度（shortTerm Scheduling）,它所调度的对象是进程（或内核级线程。）进程调度是最基本的一种调度，在多道批处理、分时和实时三种类型的OS中，都必须配置这级调度。P86

（3）中级调度（Intermediate Level Scheduling）又称中程调度（Medium-Term Scheduling）.引入中级调度的主要目的是为了提高内存利用率和系统吞吐量。

5进程的操作有哪些。

答：包括进程的创建和进程的终止

?

6进程间通信的机制有哪些。

答：消息传递系统、命名（包括直接通信和间接通信）、同步、缓冲

问答题：

1.试比较进程和程序的区别

答：进程和程序是既有联系又有区别的两个概念，它们的主要区别如下：

（1）进程是程序在处理机上的一次执行过程，是一个动态概念；而程序是代码的有序集合，其本身没有任何运行的含义，是一个静态的概念。

（2）进程是一个状态变化的过程，是有生命期的，表现在它因创建而产生，因调度 而执行，因得不到资源而暂停，因撤销而消亡；而程序是永久的，可以长久保存。

（3）进程和程序的组成不同。进程由程序、数据和进程控制块组成，而程序仅是代 码的有序集合。

（4）进程与程序之间不是一一对于的。通过多次运行，同一个程序可以对应多个进程过调用关系，一个进程可以包含多个程序。

2.并行与并发的概念

并发（Concurrent）：多个事件在同一时间段内发生。操作系统是一个并发系统，各进程间的并发，系统与应用间的并发。操作系统要完成这些并发过程的管理。

并行(parallel)是指在同一时刻发生。

1线程的基本概念及组成

答：线程，有时也被称为轻量级进程（LWP） ，是一个基本的 CPU执行单元；它包含了一个线程 ID、一个程序计数器、一个寄存器组和一个堆栈。它与属于同一个进程的其它的线程共享代码段、数据段，以及其它的操作系统资源。

?

2引入线程的好处。

答： 提高了响应速度，资源共享，经济实惠，提高了多处理机体系结构的利用率，使OS具有更好的并发性

3用户级线程和内核级线程的区别

答：对用户线程的支持通常处于内核之上，通过一个用户级线程库（thread library）实现。线程库提供了对线程的创建、调度和管理的支持，这无需来自内核的支持。用户级线程的创建和管理通常很快；

内核线程由操作系统直接支持：内核在内核空间内实现了线程的创建、调度和管理。因为线程管理由操作系统完成，所以内核线程的创建和管理要比用户线程慢。

4多线程模型有哪些，各自优缺点

多对一模型: 优点：效率比较高。缺点：如果一个线程调用了导致阻塞的系统调用的话，那么将阻塞整个进程。在多处理机环境中多个线程不能够并发执行。用户级线程库在那些采用了多对一模型不支持。

一对一模型:优点：更好的并发性；允许多个线程在多处理机环境中并行执行。缺点：在于创建一个用户线程就需要创建一个相应的内核线程。

多对多模型:优点：允许开发者随心所欲的创建用户线程。允许更大的并行性。缺点：开发者能够创建所需的用户线程，而且相应的内核线程能够在多处理机环境中并行运行。而且当一个线程执行导致阻塞的系统调用时，内核能够调度其它的线程执行。

5线程池的思想。

答：线程池的思想是在进程开始时创建一定数量的线程并将它们置入一个池（pool）中，线程在这个池中等待工作。当服务器接收到一个请求时，它就从池中唤醒一个线程（如果有可用的线程） ，由它来处理请求。一旦线程服务完毕，它就返回线程池等待后面的工作。如果池中没有可用的线程，那么服务器就等待，直到某个线程被释放。

问答题：

1.什么是线程？描述线程和进程的区别？

答：线程是在进程内用于调度和占有处理机的基本单位，它由线程控制表、存储线程上下文的用 户栈以及核心栈组成。

调度：传统操作系统中，拥有资源的基本单位和独立调度分派的基本单位都是进程；而引入线程的操作系统中，线程是调度和分派的基本单位，进程则是资源分配的基本单位。

并发性：在引入线程的OS中，进程之间可以并发执行，同一进程的多个线程之间也可以并发执行，从而使得OS具有更好的并发性。

拥有资源：在OS中，进程是拥有资源的一个独立单位，它拥有自己的资源，而线程一般不拥有系统资源，但是它可以访问其隶属进程的资源。

系统开销：创建和撤销进程涉及资源的分配或回收，需要比线程创建和撤销大得多的系统开销，同样的，进程切换的开销也远远大于线程切换的开销。

1抢占式和非抢占式区别

抢占式的：当进程从运行状态转换到就绪状态时或者当进程从等待状态转换到就绪状态时。

非抢占式的：当进程从运行状态转换到等待状态时或者当进程终止时。

在非抢占式调度下，一旦把 CPU分配给一个进程，那么该进程就会保持 CPU直到终止或转换到等待状态。

抢占式调度要付出一定的代价

2调度的准则有哪些

答：

先来先服务（FCFS）调度算法

短作业优先（SJF）或最短剩余时间优先调度算法

优先调度算法。

轮转（RR）调度算法

多级队列调度算法

多级反馈队列调度算法

要求：各种调度算法的准则

1什么是常用调度算法的评价指标？

答：CPU利用率，吞吐量，周转时间，就绪等待时间，响应时间。

吞吐量表示单位时间CPU完成作业量，周转时间指的是从作业提交到作业完结的时间间隔，就绪等待时间是每个作业在就绪队列所花的时间，响应时间是提交第一个请求到产生第一个响应的时间。

1临界区。

答：考虑由 n 个进程{P0, P1, ..., Pn- l}构成的系统。每个进程有一个代码段，被称作临界区（critical section），进程在临界区内可能会修改公有变量、更新一个表、写一个文件等等。该系统的一个重要的特征是当一个进程在其临界区内执行时就不允许其它进程在它的临界区内执行。这样，进程对临界区的执行在时间上是互斥的。

临界区是指不允许多个并发进程交叉执行，一次最多允许一个进程进入的一段程序代码。临界区是由于不同并发进程的程序段共享公 用数据或公用数据变量而引起的。这些需要互斥访问的资源称为临界区资源。

2解决临界区必须要满足的三项要求。

互斥（Mutual Exclusion） ：如果进程 Pi正在其临界区中执行，那么就不允许有其它进程在临界区中执行。

有空让进（Progress） ：如果没有进程处于临界区而此时有进程希望进入临界区，那么只可以从这些不在剩余区执行的进程中挑选出下一个进入临界区的进程，而且这个选择不可以长时间的延缓。

有限等待（Bounded Waiting） ：在一个进程请求进入临界区之后和获准之前，允许其它进程在有限的时间内进入临界区。

3信号量的定义。

答：信号量是一种同步工具。信号量 S 是一个整形数，除初始化以外，对它的访问只能通过两个标准原子操作：wait和signal。 最初， 这被称为 P操作(for wait; from the Dutch proberen, to test)和V操作(for signal; from verhogen, to increment)。

死锁的定义

答：具备一个等待队列的信号量实现可能会导致这样的一个情况：两个或多个进程无休止的等待发生一个事件，而这个事件只能由等待中的某个进程引发。问题中的这个事件是指 signal操作的执行。当达到这样的一个状态时，我们称这些进程被死锁（deadlock）

1死锁产生的四个必要条件

互斥条件：必须至少有一个资源以非共享的方式被进程持有；更确切的说，同时只有一个进程可以使用该资源。如果另一个进程请求这个资源，那么该进程必须等待这个资源被释放。

持有并等待条件：进程必须持有至少一个资源且等待获取另外的当前被其它进程持有的资源。

不可抢占条件：不可以抢占资源；也就是说，资源的释放只可以是由持有它的进程完成工作后自动释放。

循环等待条件：对一组等待进程{P0, P1, …, Pn}来说，必须：P0 等待 P1 持有的资源，P1 等待 P2持有的资源，…，Pn-1等待Pn 持有的资源，而 Pn 等待 P0 持有的资源。

2死锁的处理方法

主要有三种方法可以处理死锁：

死锁预防和死锁避免：采用某种协议预防或避免死锁，确保系统不会进入死锁状态。

死锁恢复：允许系统进入死锁状态，然后检测并恢复。

完全忽视死锁并假设系统中不会发生死锁。包括 UNIX 在内的大多数操作系统采用了这种方法。

方法2：利用信号量的保护机制实现。通过信号量mutex对eat（）之前的取左侧和右侧筷

子的操作进行保护，使之成为一个原子操作，这样可以防止死锁的出现。

伪码：

semaphore mutex = 1 ;

semaphore chopstick[5]={1，1，1，1，1};

void philosopher(int I)

{

while(true)

{

think();

wait(mutex);

wait(chopstick[(I+1)]%5);

wait(chopstick[I]);

signal(mutex);

eat();

signal(chopstick[(I+1)]%5);

signal(chopstick[I]);

}

}

1.页表的作用是实现从页号到物理块号的地址映射。

2.在页式管理系统中，用户程序中使用的地址称为 逻辑地址 ，实际访问主存时由系统将它转化为 物理地址 。

3.分页管理是把内存分为大小相等的区，每个区称为页帧(或页框)，而把程序的逻辑空间分为若干页，页的大小与页帧的大小 相等 。

4.在分页存储管理中，为了加快地址变换速度，页面大小的值常取2的整数次幂。

5.在请求式分页系统中，被调出的页面又立刻被调入，这种频繁的调页现象称为颠簸。

6.分段管理中，若逻辑地址中的段内地址大于段表中该段的段长，则发生 地址越界中断。

7.段页式存储管理中，每道程序都有一个 段 表和若干个 页 表。

8.页式管理系统的逻辑地址结构由 页号 和 页内位移 组成。

9．分段管理中的地址映射过程是：首先找到该作业段表的 起始地址 ，然后根据逻辑地址中的 段号 去查找段表得到该段的内存起始地址，再与逻辑地址中的 段内位移 相加得到物理地址。

10.请求分页存储管理也称为动态页面管理，不是把一个进程映象的所有页面一次性全部装入内存，而只装入一部分，其余部分在执行中动态调入。

11.在段页式管理中，逻辑地址分解为段号、页号、页内位移 三部分。

1. 为什么要引入动态重定位？如何实现？

答：

（1）系统在内存管理中经常需要将进程浮动，以整理出较大的存储空间。为了适应进程的这种地址变化，需要对进程的地址进行变换，即动态重定位。

（2）硬件上设置“重定位寄存器”，专门存放进程的首地址。程序执行时的内存物理地址是由重定位寄存器中的地址和相对地址相加得到的。当进程从内存的某处移动到另一处时，不需对程序做任何修改，只要将进程的新地址替换原来的旧地址即可。

2.试比较分段式和分页式存储管理方式的主要差别。

答：它们的差别主要表现在以下几个方面：

（1）页面是信息的物理单位，分页是为了实现非连续分配，以便解决内存碎片问题，或者说分页是由于系统管理的需要。段是信息的逻辑单位，它含有一组意义相对完整的信息，分段的目的是为了更好地实现共享，满足用户的需要。

（2）页面的大小固定且由系统确定，将逻辑地址划分为页号和页内地址是由机器硬件实现的。而段的长度却不固定，它取决于用户所编写的程序，通常由编译程序在对源程序进行编译时根据信息的性质来划分。

（3）分页式存储管理的作业地址空间是一维的，分段式存储管理的作业地址空间是二维的。

1.什么是颠簸？产生颠簸的原因是什么？

答：

（1）颠簸是由于内存空间竞争引起的。当需要将一个新页面调入内存时，因内存空间紧张，不得不将一个旧页面置换出去，而刚刚置换出去的旧页面可能又要被使用，因此需要重新将它调入。若一个进程频繁地进行页面调入调出，势必加大系统的开销，使系统运行效率降低。通常称这种现象为该进程发生了颠簸。

（2）产生颠簸的原因主要有：系统内的进程数量太多，致使一个进程分得的存储块过少；系统采取的置换算法不够合理。

2.常见的页面置换算法

答：最佳页面置换算法(OPTIMAL)、先进先出页面置换算法(FIFO)、最近最久未用置换算法(LRU)、LFU置换算法

最佳页面置换算法(OPTIMAL)：所选择的被淘汰页面，将是以后永不使用的，或许是在最长（未来）时间内不再被访问的页面。采用最佳置换算法，通常可保证获得最低的缺页率。

先进先出页面置换算法(FIFO)：总是淘汰最先进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰。

最近最久未用置换算法(LRU)：选择最近最久未使用的页面予以淘汰。

LFU置换算法：选择在最近时期使用最少的页面作为淘汰页。

3.缺页的概念，页表的含义

缺页：要访问的页面不在主存，需要操作系统将其调入主存后再进行访问。

页表：用来将虚拟地址空间映射到物理地址空间的数据结构称为页表。

4.实现虚拟存储器需要哪些硬件支持

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；

1. 文件管理有哪些主要功能？其主要任务是什么？

答：文件管理的主要功能和主要任务有以下四个方面：

（1）外存空间管理。其主要任务是为每个文件分配必要的外存空间，提高外存的利用率，并能有助于提高文件系统的效率。

（2）目录管理。其主要任务是为每个文件建立目录项，并对众多的目录项加以有效的组织，以实现方便的按名存取操作。

（3）文件读写操作。其主要任务是根据用户请求从外存中读取数据，或将数据写入外存。

（4）存取权限控制。其主要任务是防止未经核准的用户访问文件；防止冒名顶替存取文件；防止以不正确的方式访问文件。

1.磁盘调度算法有哪些？每种方法的优缺点。

答:FCFS、SSTF、扫描(SCAN)算法 、循环扫描(CSCAN)算法

FCFS：先来先服务，它根据进程请求访问磁盘的先后次序进行调度。

SCAN:扫描算法，磁头不停的往复运动，由边缘至中心然后返回，沿途执行已经到来的访问。

CSCAN：循环扫描算法，在SCAN算法的基础上规定磁头单向移动。

1.有哪几种I/O控制方式？

答：程序I/O方式、中断举动I/O控制方式、直接存储器访问（DMA）I/O控制方式和I/O通道控制方式。

2.设备管理的主要功能和主要任务

答：

主要功能：缓冲管理，设备分配和设备处理，以及虚拟设备等.

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.

● 缓冲管理：提高CPU的利用率进而提高系统的吞吐量

● 设备分配：根据用户进程的I/O请求、系统的现有资源以及按照某种设备的分配策略，为之分配其所需的设备

● 设备处理：用于实现CPU和设备控制器之间的通信

3. 设备分配时应考虑的因素

答：设备的固定属性、设备分配算法、设备分配时的安全性、设备独立性

（1）设备的固有属性有3种：  
独占性：设备在一段时间内只允许一个进程独占，eg：临界资源  
 共享性：设备允许多个进程同时共享  
 可虚拟设备：设备本身随时独占设备，但经过某种技术处理，可以把它改造成虚拟设备  
（2）设备分配算法：先来先服务、优先级高者优先  
（3）设备分配中的安全性：安全分配方式、不安全分配方式

4. 为什么引入缓冲（目的是什么？）

答：在设备管理中，引入缓冲区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cpu中断响应时间的限制

（3） 提高cpu和I/O设备之间的并行性

试从调度性，并发性，拥有资源和系统开销几个方面对线程与进程进行比较

1)调度  
● 在传统的操作系统中，作为拥有资源的基本单位和独立调度、分派的基本单位都是进程。  
● 在引入线程的操作系统中，把线程作为调度和分派的基本单位，而进程作为资源拥有的基本单位，把传统进程的两个属性分开，使线程基本上不拥有资源，这样线程便能轻装前进，从而可显著地提高系统的并发程度。  
● 在同一进程中，线程的切换不会引起进程的切换，但从一个进程中的线程切换到另一个进程中的线程时，将会引起进程的切换。

2)并发性  
在引入线程的操作系统中，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，使得操作系统具有更好的并发性，从而能更加有效地提高系统资源的利用率和系统的吞吐量。

3) 拥有资源  
 ● 不论是传统的操作系统，还是引入了线程的操作系统，进程都可以拥有资源，是系统中拥有资源的一个基本单位。  
● 一般而言，线程自己不拥有系统资源(也有一点必不可少的资源)，但它可以访问其隶属进程的资源，即一个进程的代码段、数据段及所拥有的系统资源，如已打开的文件、I/O设备等，可以供该进程中的所有线程所共享。

4) 系统开销  
● 在创建或撤消进程时，系统都要为之创建和回收进程控制块，分配或回收资源，如内存空间和I/O设备等，操作系统所付出的开销明显大于线程创建或撤消时的开销。  
● 就切换代价而言，进程也是远高于线程的。此外，由于一个进程中的多个线程具有相同的地址空间，在同步和通信的实现方面线程也比进程容易。在一些操作系统中，线程的切换、同步和通信都无须操作系统内核的干预。