一． 操作系统

操作系统：是控制和管理计算机系统的硬件和软件资源，合理的组织计算机工作流程以及方便用户使用,是一种系统软件。
操作系统作用:设置操作系统的目的就是提高计算机系统的效率，增强系统的处理能力，充分发挥系统的利用率，方便用户使用。
硬件基础---中断和通道
中断 指 CPU 在收到外部中断信号后，停止原来工作，转去处理该中断事件，完毕后回到原来断点继续工作。
中断过程：中断请求，中断响应，中断点（暂停当前任务并保存现场），中断处理例程，中断返回（恢复中断点的现场并继续原有任务）
通道(又称为 I/O 处理机) 实际上是一台功能单一、结构简单的 I/O 处理机，它单独与 CPU ，并直接控制外部设备，与内存进行数据传输。
【CPU 与通道的通讯】：CPU 与通道之间为主（CPU ）从关系（通道），采用通道进行数据传输的过程如下：CPU 向通道发出 I/O 指令；通道执行通道程序进行 I/O 操作；I/O 完成或出错时，以中断方式请求 CPU 处理。
多道程序设计技术：在计算机内存中同时存放几道相互独立的程序，它们在管理程序的控制下相互穿插地运行，共享 CPU 和外设等资源。采用多道程序设计技术的批处理系统称为多道批处理系统
操作系统是一种系统软件，在操作系统中采用多道程序设计方式能提高 CPU 和外部设备的利用效率。一般来说，为了实现多道程序设计，计算机需要有更大的内存。
三种操作系统基本类型批处理系统、分时系统、实时系统
分时系统：多个用户分时（按时间划分轮流）的使用同一计算机的系统称为为分时系统。
分时系统中，为使多个用户能够同时与系统交互，最关键的问题是 能在一短的时间内，使所有用户程序都能运行

分时特点：

- 同时性或多路性：多用户同时操作、使用计算机
- 独占性：各终端用户感觉到自己独占了计算机；
- 及时性：用户的请求能在较短时间内相应；
- 交互性：用户能与计算机进行人——机对话。

响应时间为用户发出一条指令到系统处理完这条指令并做出回答所需要的时间

实时操作系统主要用于过程控制、事务处理等有实时要求的领域，其主要特征是**实时性和可靠性**

在设计分时操作系统时，首先要考虑的是 交互性和响应时间 ；在设计实时操作系统时，首先要考虑的是 实时性和可靠性 ；在设计批处理系统时，首先要考虑的是周转时间和系统吞吐量

目前的操作系统，通常具有分时、实时和批处理功能，又称作通用操作系统。

现代操作系统**主要特征**：

1）**并发性**

单处理机、多道程序处理时，宏观上并发，微观上交替执行。并发指的是进程，操作系统是一个并发系统。

2）**共享性**

多个进程共享有限的计算机系统资源，系统合理分配，资源在一个时间段内交替被多个进程所用。

3）**虚拟性**

一个物理实体映射为若干个对应的逻辑实体（分时或分空间）。虚拟是操作系统管理系统资源的重要手段，可提高资源利用率。

4）**异步性**

异步性也称不确定性，指进程的执行顺序和执行时间及执行结果的不确定性：

A 程序执行结果不确定，不可再现

B 多道程序设计环境下，程序按异步方式运行。

操作系统**主要功能**：

1）处理机管理：

2）存储管理

3）设备管理

4）信息管理

5）用户接口

二． 接口

作业：用户在一次解题过程中或一个事务处理中要求计算机系统所作工作的总和，它是用户向计算机系统提交一项工作的基本单位 。

作业步：是在一个作业的处理过程中，计算机所做的相对独立的工作。

作业流：批量系统中需要将一批作业依次输入到辅助存储器中，形成作业流。

作业类型：

- 脱机作业：也称为批量型操作，在一次业务处理过程中，从输入程序和数据到输出结果的全过程。
- 联机作业：也称为交互型操作或终端操作，是指用户直接与计算机系统交互作用来控制作业的运行，多出现在分时系统和单用户微机操作系统中。**作业调度算法**

作业调度算法：

- FCFS：先来先服务算法，谁先来就等到资源足够时执行谁。
- SJF：短作业优先，优先执行时间短的作业执行，但是会导致先到的耗时作业一直不执行，降低用户体验
- HRF：高响应比优先算法：等待时间/计算时间。缺点：作业多时，计算响应时间耗时
- 优先级调度算法：有两种方式：一是用户设定优先级，二是系统设定优先级，优先级高的先执行
- 均衡调度算法：根据使用不同的资源进行分类，使得占用不同资源的作业能够同时执行，从而提高效率，和系统的吞吐量。

三． 进程

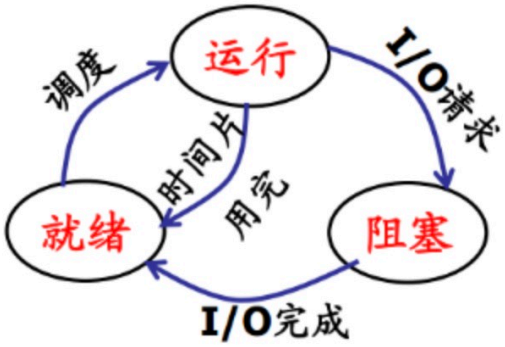
进程的表示 程序 数据 进程控制块 PCB（process control block）
进程的特征：

- 结构特征
- 动态性 由创建产生，由调度执行，由撤销消亡
- 并发性
- 独立性 进程实体是一个能独立运行、独立分配、和独立接受和调度基本单位
- 异步性 各进程以各自独立，不可预知的速度向前推进

进程与程序的区别

- 进程是程序的一次执行，该程序可以与其它程序并发执行。
- 进程是动态的，程序是静态的：程序是有序代码的集合；进程是程序的执行。通常进程不可在计算机之间迁移；而程序通常对应着文件、静态和可以复制。
- 进程是暂时的，程序是永久的：进程是一个状态变化的过程，程序可长久保存
- 进程与程序的对应关系 :通过多次执行,一个程序可对应多个进程 ;通过调用关系,一个进程可包括多个程序。
- 进程与程序的组成不同：进程的组成包括程序、数据和进程控制块（即进程状态信息）

进程状态：



原子操作：由若干条机器指令构成的并用以完成特定功能 的一段程序，而且这段程序在执行期间不允许中断。
原语（原子操作）和广义指令的区别：
（1）原语的执行是不可分割的，而广义指令所包含的程序段是允许被中断的，不要求具有不可分割性。
（2）广义指令的功能可以在用户态下实现，而原语只能在系统态下执行。
进程的中止（撤销）：正常结束、异常结束、外界干预

进程与线程

- 线程是进程的一个组成部分。每个进程创建时通常只有一个线程，需要时可创建其他线程。
- 进程的多线程都在进程的地址空间活动。
- 资源是分给进程的，不是分给线程的。线程在执行中需要资源时，可从进程资源中划分。
- 处理机调度的基本单位是线程，线程之间竞争处理机。真正在 CPU 上运行的是线

程。

- 线程在执行时，需要同步。

进程的基本属性：

（1）进程既是一个拥有资源的独立单位，它可独立分配虚地址空间、主存和其它系统资源；

（2）进程又是一个可独立调度和分派的基本单位。

线程：进程内一个执行单元或一个可调度实体

资源拥有单元称为进程（或任务），调度的单位称为线程。在具有多线程的操作系统中，处理机调度的基本单位是线程。一个进程可以有多个线程，而且至少有一个可执行线程

死锁：两个或两个以上并发进程，如果每个进程持有某种资源，而又等待着别的进程释放它或它们现在保持着的资源，否则就不能向前推进。此时，每个进程都占用了一定的资源，但又都不能向前推进。这种现象称为死锁（所有进程的申请都未得到满足，都在等待别的进程释放）

死锁产生的必要条件：

必须具备四个必要条件才会发生死锁

- 互斥条件：一个资源每次只能给一个进程使用。
- 不可剥夺条件：资源申请者不能强行的从资源占有者手中夺取资源,资源只能由占有者自愿释放。
- 请求和保持条件：在申请新的资源的同时保持对原有资源的占有。
- 循环等待条件：存在一个进程-等待资源环形链

处理死锁的办法：

1）鸵鸟策略：

采用不理睬策略

2）预防策略：

破坏产生死锁的四个必要条件。

3）避免策略：

精心的分配资源，动态的回避死锁。

4）检测和解除：

发生死锁后及时能检测出，并还能采取措施解除。

预防死锁的办法：

预防死锁的方法是破坏产生死锁的四个必要条件之一。

1）破坏互斥条件

互斥使用是资源本身特征所决定的。使用硬软件结合可改变资源本身特性，例如采用SPOOLing 技术可将 “独享” 打印机改变为“共享” 的打印机。

2）破坏不可剥夺条件

一个进程在申请新的资源不能立即满足而变为阻塞状态之前，必须释放已占有的全部资源

3）破坏请求和保持条件——资源静态预分配

在运行前，一次性将其所需要的所有资源分配给该进程。

4）循环等待条件 ——有序资源使用法把系统中的全部资源分别分给一个特定的序号，并且要求每个进程均应严格地按照序号递增的次序请求资源，否则操作系统不予分配。

死锁的**避免**：

该方法允许进程动态地申请资源，系统在进行资源分配之前，先计算资源分配的安全性。避免死锁的实质是使系统不进入不安全状态

死锁的**检测**：
系统的状态可以用 资源分配图 来描述，可用资源分配图简化来判断系统是否处于死锁状态
死锁定理（死锁状态的充分条件）当且仅当系统某状态 S 所对应的 资源分配图 是不可化简的，则 S 是死锁状态。
死锁的解除：
1）撤消进程：强制性地从系统中撤消进程并剥夺它们的资源给剩下的进程使用：
A.进程的优先数；
B.重新启动它并运行到当前撤消点所需的代价；
C.作业的外部代价：即与此进程相关的作业类型都可以有其相应的固定撤消代价。
2）剥夺资源：挂起和解挂机构:
从被挂起进程那里强占资源以解除死锁
产生死锁的基本原因是 资源分配不当 和 进程推进顺序非法

PV 操作，信号量机制

四． 存储管理

虚拟存储器:虚拟存储器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加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。
虚拟存储器的实现方法：请求分页系统
请求分页的页表机制、缺页中断机构、地址变换机构
地址映射就是要建立虚拟地址与内存地址的关系：静态分配、动态分配
静态要求是非连续存储的，动态可以是连续存储的，动态存储是虚拟存储的基础。
为了实现扩容，程序执行过程中必须内存外存经常交换数据。

动态分区的分配和回收;
找合适---更新---合并
基于顺序搜索的动态分配分区算法：
一、首次/最先适应算法（First Fit）：---按起始地址递增的次序排列。
特点： 该算法倾向于使用内存中低地址部分的空闲区，在高地址部分的空闲区很少被利用，从而保留了高地址部分的大空闲区。显然为以后到达的大作业分配大的内存空间创造了条件。
缺点：低地址部分不断被划分，留下许多难以利用、很小的空闲区，碎片化，而每次查找又都从低地址部分开始，会增加查找的开销。
二、最佳适应算法（Best Fit）：---空闲区按从小到大排。
特点：每次分配给文件的都是最合适该文件大小的分区。
缺点：内存中留下许多难以利用的小的空闲区碎片。
三、最坏适应算法(Worst Fit)：---按空闲区从大到小排。
特点：给文件分配分区后剩下的空闲区不至于太小，产生碎片的几率最小，对中小型文件分配分区操作有利。
缺点：使存储器中缺乏大的空闲区，对大型文件的分区分配不利。
从搜索速度上看，最先适应算法具有最佳性能。从回收过程来看，最先适应算法也是最佳的。

最佳适应算法找到的空闲区是最佳的。
最坏适应算法 基于不留下碎片空闲区这一出发点的。
动态分区时的回收与别的空闲区进行拼接，起始地址是上空闲区起始地址。

页式管理为了减少碎片化，页式管理页内连续但是页与页不连续,使用存储页面表
抖动现象：置换算法选择不当，有可能产生刚被调出内存的页又要马上被调回内存，调回内存不久又马上被调出内存，如此反复的局面。这使得整个系统的页面调度非常频繁，以致大部分时间都花费在主存和辅存之间的来回调入调出上
最不经常使用 LFU(least frequently used) 0/1
该算法在需要淘汰某一页时，淘汰到当前时间为止，被访问次数最少的那一页。在页表中给每一页增设一个访问计数器
理想型淘汰算法 OPT(optimalreplacement algorithm))
算法淘汰在访问串中将来再也不出现的或是在离当前最远的位置上出现的页。
最近最久未使用页面置换算法(least recently used))
该算法的基本思想是: 当需要淘汰某一页时，选择离当前时间最近的一段时间内最久没有使用过的页先淘汰
FIFO 算法认为先调入内存的页不再被访问的可能性要比其他页大，因而选择最先调入内存的页换出。
页式管理的优缺点：
有效地解决了碎片问题，提高了主存的利用率，又有利于组织多道程序执行
要求相应的硬件支持，增加了系统开销，算法不当可能会发生抖动，最后一页损失
段式管理的淘汰算法--
FIFO,LRU,最佳

五． 设备

通道是一个独立于 CPU 的专管输入输出控制的处理机，它控制设备与内存直接进行数据交换。它有自己的通道指令，这些通道指令受 CPU 启动，并在操作结束时向 CPU 发中断信号。
数据传送控制方式有哪几种? 试比较它们各自的优缺点
外围设备和内存之间常用数据传送控制方式

- 程序直接控制方式：控制简单，硬件支持少；但是只能 CPU 和设备串行工作，，CPU 利用率低，不能是实现设备并行。
- 中断控制方式：硬件要求少，提高了 CPU 利用率但是小号的 CPU 处理时间多。
- DMA 方式：I/O 速度快,减轻 CPU 中断次数，排除数据丢失等现象；缺点所需硬件多，多个 DMA 容易发生内存地址冲突。
- 通道方式：I/O 速度快,减轻 CPU 的工作负担和增加了计算机并行能力；单控制较复杂，所需硬件复杂。

数据的组织和格式：存储面 扇区 磁道
磁盘调度

- 先来先服务 FCFS：公平、简单，且每个进程的请求都能依次地得到处理。未对寻道进行优化，致使平均寻道时间可能较长。
- 最短寻道时间优先(SSTF)：磁道与当前磁头所在的磁道距离最近。以使每次的寻道时间最短。但不能保证平均寻道时间最短。可能导致某个进程发生“饥饿”(Starvation)现象。

- 扫描(SCAN) 算法 :磁道与当前磁道间的距离，与磁头移动的方向----电梯式上下。可防止老进程出现 “饥饿” 现象，广泛用于大、中、小型机器和网络中的磁盘调度。
- 循环扫描(CSCAN)算法：磁道与当前磁道间的距离，磁头单向移动---0 到 100，又 0 到 100

六． 文件资源

文件系统：操作系统中与管理文件有关的软件和数据称为文件系统。
文件的分类
【按性质用途分】:
- 系统文件
- 库文件
- 用户文件
- 【按组织形式分】:
- 普通文件
- 目录文件
- 特殊文件

【按信息流向分】:
- 输入文件
- 输出文件
- 输入/ 输出文件

【按保护级别分】:
- 只读文件
- 读写文件
- 可执行文件
- 不保护文件

文件的分类的目的 主要是为了提高处理速度 和 起保护与共享 的作业。

常见面试题

一．进程与线程的关系以及区别

1.定义：

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位.

线程是进程的一个实体,是 CPU 调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源.

2.关系：

一个线程可以创建和撤销另一个线程;同一个进程中的多个线程之间可以并发执行.

相对进程而言，线程是一个更加接近于执行体的概念，它可以与同进程中的其他线程共享数据，但拥有自己的栈空间，拥有独立的执行序列。

3. 区别：

比较	进程	线程
活泼性	不活泼（只是线程的容器）	活泼
地址空间	系统赋予的独立的虚拟地址空间（对于32位进程来说，这个地址空间是4GB）	在进程的地址空间执行代码。线程只有一个内核对象和一个堆栈，保留的记录很少，因此所需要的内存也很少。因为线程需要的开销比进程少
调度	仅是资源分配的基本单位	独立调度、分派的基本单位
并发性	仅进程间并发（传统OS）	进程间、线程间并发
拥有资源	资源拥有的基本单位	基本上不拥有资源
系统开销	创建、撤销、切换开销大	仅保存少量寄存器内容，开销小。

二．Windows 下的内存是如何管理的

1.虚拟内存：

最适合用来管理大型对象或者结构数组

2.内存映射文件：

最适合用来管理大型数据流（通常来自文件）以及在单个计算机上运行多个进程之间共享数据

3.内存堆栈：

最适合用来管理大量的小对象

三．中断和轮询的特点

轮询：定时对各种设备轮流询问一遍有无处理要求。轮流询问之后，有要求的就加以处理。

特点：轮询效率低，等待时间很长，CPU 利用率不高；中断容易遗漏一些问题，CPU 利用率高。

四．什么是临界区，如何解决冲突

1. 每个进程中访问临界资源的那段程序称为临界区，每次只准许一个进程进入临界区，进入后不允许其他进程进入。
2. 进入临界区的进程要在有限时间内退出，以便其他进程能及时进入自己的临界区。
3. 如果不能进入自己的临界区，就应该让出 CPU，避免进程出现忙等等现象。

五．分段与分页的区别

页是信息的物理单位，分页是为了实现离散分配方式，以减少内存的外零头，提高内存的利用率。分页仅仅是由于系统管理的需要，而不是用户的需要。

段是信息的逻辑单位，它含有一组其意义相对完整的信息。分段的目的是为了能更好的满足用户的需要。

区别：页的大小固定且由系统确定，把逻辑地址分为页号和页内地址两部分，由机器硬件实现的。因此一个系统只能有一种大小的页面。段的长度却不固定，决定于用户所编写的程序，通常由编写程序在对源代码进行编辑时，根据信息的性质来划分。

六．进程间通信方式

1. 管道（pipe）：管道是一种半双工的通信方式，数据只能单向流动，而且只能在有血缘关系的进程间使用，进程的血缘关系通常是指父子进程关系。
2. 命名管道（named pipe）：也是半双工的通信方式，但是它允许无亲缘关系关系进程间通信。
3. 信号（signal）：是一种比较复杂的通信方式，用于通知接收进程某一事件已经发生。
4. 信号量（semophere）：信号量是一个计数器，可用来控制多个进程对共享资源的访问。它通常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。
5. 消息队列（message queue）：消息队列是由消息组成的链表，存放在内核中，并由消息队列标识符标识。消息队列克服了信号传递消息少，管道只能承载无格式字节流以及缓冲区大小受限等缺点。
6. 共享内存（shared memory）：就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问，共享内存是最快的 IPC 方式，它是针对其他进程间的通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号量等配合使用，来实现进程间的同步和通信。
7. 套接字（socket）：套接口也是进程间的通信机制，与其他通信机制不同的是它可用于不同及其间的进程通信。

几种方式的比较：

管道：速度慢、容量有限

消息队列：容量收到系统限制，且要注意第一次读的时候，要考虑上一次没有读完数据的问题。

信号量：不能传递复杂信息，只能用来同步。

共享内存：能够很容易控制容量，速度快，但要保持同步，比如一个进程在写的时候，另一个进程要注意读写的问题，相当于线程中的线程安全。

七．线程间的通讯机制

1.锁机制：互斥锁、条件变量、读写锁

互斥锁提供了以排他方式防止数据结构被并发修改的方法。

读写锁允许多个线程同时读共享数据，而对写操作是互斥的。

条件变量可以以原子的方式进行阻塞进程，直到某个特定条件为真为止。对条件的测试是在互斥锁的保护下进行的。条件变量始终与互斥锁一起使用。

2.信号量机制：包括无名信号量和命名线程信号量
3.信号机制：类似进程间的信号处理
线程间的通信目的主要是用于线程同步，所以线程没有像进程通信中的用于数据交换的通信机制。

八．死锁？产生条件？如何避免？

见上面。

九．进程间同步与互斥的区别，线程同步的方式？

互斥：指某一个资源同时只允许一个访问者对其进行访问，具有唯一性和排它性。但互斥无法限制访问者对资源的访问顺序，即访问是无序的
同步：是指在互斥的基础上（大多数情况下），通过其它机制实现访问者对资源的有序访问。大多数情况下，同步已经实现了互斥，特别是所有写入资源的情况必定是互斥的。少数情况是指可以允许多个访问者同时访问资源。
同步：体现的是一种协作性。互斥：体现的是排它性。
进程同步的主要任务：是对多个相关进程在执行次序上进行协调，以使并发执行的诸进程之间能有效地共享资源和相互合作。从而使程序的执行具有可再现性。
进程同步机制遵循的原则：

1.空闲让进；
2.忙则等待；
3.有限等待；
4.让权等待；

线程同步是指多个线程同时访问某资源时，采用一系列的机制以保证最多只能一个线程访问该资源。