

1. 操作系统的特点和功能是什么？

答：操作系统的特点是并发、共享、虚拟、异步。其中，**并发和共享是操作系统主要的特点**。操作系统的功能主要包括：处理机管理、存储器管理、设备管理和文件管理等。操作系统管理计算机的全部软、硬件资源，合理组织计算机的工作流程，通过虚拟、并发等技术提升计算机计算效率，提升资源利用率。

2. 中断与系统调用的区别？

答：中断是CPU对例如I/O设备发送的中断信号的响应，此时CPU应该检测当前是否有正在处理的程序，如果有，则需要保存CPU上下文环境，从用户态切换至内核态对中断请求进行处理，完成处理后再返回原处理程序的断点位置继续执行。

中断通常分为硬中断和软中断两种。硬中断是指从硬件发出的中断信号，例如I/O请求。而软中断是指由指令执行过程中发出的中断。系统调用就是一种软中断，它是指通过应用程序通过操作系统间接地调用I/O过程。由于应用程序只能执行于用户态，而I/O操作需要进入内核态，因此通过中断请求，操作系统进行用户态向内核态的转换完成I/O操作。

3. 进程和线程的概念以及区别是什么？

答：进程是指在多道程序环境下能够并发执行的活动中程序的描述，进程是操作系统资源分配的基本单位。而线程是进程的一个实体，是操作系统调度的基本单位，只拥有少部分的资源，多个从属于同一个进程的线程将共享该进程的资源。

进程和线程的区别可以从以下几点进行描述：

- ① 进程和线程都具有调度性与并发性；
- ② 进程是拥有资源的基本单位，而线程除了少部分运行必要的资源之外，基本不拥有系统资源，从属于一个进程的多个线程将共享该进程的资源；
- ③ 在独立性上，同一个进程中的不同线程之间的独立性比不同进程之间的独立性差，因为多个线程将共享进程资源，通常其执行会相互约束；
- ④ 在系统开销上，进程的创建、切换与撤销的系统开销大于线程，切换进程需要切换上下文环境，而线程的切换只需要更改少部分寄存器的内容；
- ⑤ 在多处理器环境中，线程有更大的优势，将同一进程的不同线程分配于不同处理器上可以加速程序的执行。

4. 线程共享和不共享的资源有哪些？

答：线程共享的资源有：堆、全局变量、静态变量、文件等；

线程不共享的资源有：栈、寄存器。

5. 进程有哪几种状态？他们之间如何进行转换？

答：进程的基本转态有：**就绪、执行、阻塞**。就绪状态：进程已经分配到除了处理机之外所有的执行所需要的资源，一旦获得处理器就可以立刻执行；执行状态：进程获得处理机并且在处理器上执行；阻塞状态：当执行中的程序由于等待某个事件发生而无法执行时，放弃处理机从而进入阻塞状态，例如等待I/O完成。三种状态之间的相互转换：执行状态的程序遇到I/O请求等将会转入阻塞状态，I/O请求完成时会进入就绪状态，获得处理机之后将重新进入执行状态。执行状态的程序当时间片用完时，将会重新回到就绪状态。

Q：为什么没有就绪→阻塞和阻塞→执行：就绪状态的进程由于没有获得处理机，因此其状态不会改变；阻塞状态的进程被唤醒后需要先进入就绪度列，才会被调度程序选中进入执行状态。

6. 进程的调度策略有哪些？

答：① **先来先服务**：这是最简单的调度算法，使用于作业调度和进程调度。在作业调度中，该算法每次会从后备**队列** 中选择最先进入该队列的作业调入内存，为其分配资源并且创建进程，并放入就绪队列；在进程调度中，算法将从就绪队列中选择最先进入队列的进程，分配处理机执行；

② **短作业优先**：顾名思义，在进行进程或者作业调度之前，操作系统将获取后备队列中作业的执行时间或者就绪队列中进程的执行时间，并且选择时间最短的优先进行调度。该算法会导致长作业或者长进程的饥饿现象；

③ **优先权调度**：为了照顾一些紧迫类型的作业，为每个作业按照紧急程度赋予了优先级。在进行作业调度时，系统将从**后备队列**中选择拥有高优先权的作业装入内存。该算法可以分为**抢占式**和**非抢占式**，在抢占式方式下，一旦在程序执行期间有比当前执行程序优先权更高的进程进入就绪队列，调度程序将立即停止当前执行程序，转而将处理机分配给高优先权的进程；而在非抢占方式下，执行的程序会一直执行直至完成或者主动放弃处理机，调度程序才进行调度。

④ **高响应比优先调度**：为了解决短作业优先算法中长作业的饥饿问题，该算法为每个作业引入一个动态优先权，该优先权等于（等待时间 + 要求服务时间）/ 要求服务时间，该算法在利用短作业优先的思想之下，同时照顾等待时间长的作业（周转时间/要求服务时间）；

⑤ **轮转算法**：系统将所有就绪进程排成一个队列，在每次调度时，处理机将分配给队列的起一个进程，并且令其执行一个时间片长度。当时间片执行完时，由计时器发出一个时钟请求，此时调度程序停止该程序并将其移动至队尾；

⑥ **多级反馈队列调度**：该算法不需要知道各种进程的执行时间，同时可以满足各种不同类型进程的需求，例如短进程会很快完成，长进程不会饥饿。算法的执行过程为：

- （1）设置多个就绪队列，每个队列的优先级不同。第一个队列优先级最高并依次递减，优先级越低的队列时间片越长，一般按照倍数增长；
- （2）当一个新的进程进入内存后，其首先被放入第一队列的队尾，并且按照先来先服务的原则等待调度。若其在第一个时间片可以完成执行，则将撤出系统；若不能，则其在时间片执行完毕之后会被移动至下一个就绪队列的队尾，如此反复；
- （3）上一级队列为空的时候，下一级队列才开始进行调度。

7. 进程间的通信方式有什么？

答：进程间的通信方式主要有以下几种：① **共享存储器系统**：相互通信的进程共享某些数据结构或者共享存储区，并且借助这些空间进行通信，主要有基于共享数据结构的通信方式（例如队列实现的缓冲区）与基于共享存储区的通信方式（例如内存中划分出来的共享区）；② **管道通信系统**：管道是连接一个读进程与一个写进程之间通信的一个共享文件，又称为pipe文件，它必须被互斥地访问，即运作于半双工方式。同时进行同步，即一方写满后另一方才读，或者一方读空后另一方才写；③ **消息传递系统**：在该机制中，以格式化的消息（message）为单位，将通信的数据封装在消息中，并利用操作系统提供的一组通信命令（原语）在进程之间进行数据交换；④ **客户机-服务器系统**：该机制是在网络环境中主流的通信机制，主要分为套接字、远程过程调用和远程方法调用。

8. 什么是死锁？产生死锁的原因？产生死锁的必要条件是什么？

答：**死锁是指多个进程因为竞争资源而造成相互等待的情况**，若无外力作用，这些进程都无法向前推进。

产生死锁的原因有：① **竞争不可抢占型资源**；② **竞争可消耗型资源（硬件中断、信号等）**；③ **进程推进顺序不当**。

产生死锁的四个必要条件：

- ① **互斥条件**：进程对所获得的资源必须互斥访问，即同一时刻只能有一个进程访问；
- ② **请求和保持**：进程在至少获得一个资源的情况下，又提出新的资源需求；

③ 不可抢占：进程所获得的资源在其完成之前不可被其他进程抢占；

④ 循环等待链：当发生死锁时，必然存在一个进程——资源循环等待链。

9. 如何预防死锁？

答：预防死锁可以通过破坏产生死锁的四个必要条件中的一个实现。① 破坏请求保持条件：资源一次性分配，则一次性分配进程运行所需要资源，或分配进行运行初期需要的资源，在进程运行过程中逐步释放资源并且请求新资源；② 破坏不可抢占条件：当进程获得部分资源，但是得不到其他资源时，它将释放已经占有的资源；③ 破坏循环等待链：资源有序分配，即系统为没类资源赋予一个编号，每一个进程按照编号递增的顺序请求资源。

10. 什么是银行家算法？

答：银行家算法是一种用于避免死锁的算法，该算法将避免系统进入一种不安全状态，即可能会产生死锁的状态。其允许进程动态地申请资源，但在系统进行资源分配之前，应该首先计算此次分配资源的安全性，如果分配不会导致系统进入不安全状态则分配，否则等待。

银行家算法的实现过程为：设置四种数据结构：Available（系统中所有可用资源数量）、Max（系统中n个进程对m类资源的最大需求量）、Allocation（每类资源已经分配给进程的資源数）、Need（每个进程尚需的各类资源数）。申请过程中，Requires需要小于Need与Available，若满足，则系统试探性的进行分配，并执行安全性算法，假设将可用资源分配给某一进程，进程完成后释放资源。如果所有进程可以完成则说明系统仍处于安全状态，则此次分配成功。否则，不进行分配并且等待。

11. 哲学家进餐有哪些实现方式？

答：① 最多只允许n-1个哲学家拿起筷子就餐；② 奇数号哲学家先拿左边的筷子，偶数号哲学家先拿右边的筷子；③ 当一个哲学家拿起一只筷子时，如果另外一只筷子无法拿起，则放下刚刚拿起的筷子；④ 一次只允许一名哲学家就餐，待其就餐完毕后才允许其他哲学家就餐。

12. 分页的作用于好处是什么？分页与分段有什么区别？

答：分页的作用是为了实现分连续分配，从而解决内存碎片的问题，从而提高内存利用率，便于系统管理。其中页面的大小由系统决定，将内存的逻辑地址划分为页号与页内偏移，并由硬件地址转换机构配合页表实现从逻辑地址到物理地址的映射。

而段是信息的逻辑单位，它含有一组意义相对完整的信息，分段的目的则是为了更好地实现共享从而满足用户的需要，同时避免逻辑地址中大量的内存浪费。段的大小不固定，而取决于用户所编写的程序。分页的作业地址是一维，而分段是二维，需要同时提供段名的段内地址。

13. 内存分配机制有哪些？

答：以Java中的JVM为例，其内存分区通常被划分为五个区域：程序计数器、本地方法栈、方法区、栈、堆。

程序计数器：每个正在执行的线程都可以获得自己的程序计数器PC，PC始终指向下一条即将被执行的指令；

栈（堆栈）：每个新创建的线程都会分配一个栈，JVM对栈进行压栈与出栈的操作。当线程激活一个Java方法，JVM就会在堆栈中压入一个帧，用于保存执行此方法期间的参数、局部变量。栈的存取速度快，并且栈数据可以进行共享；

本地方法栈：用于存储本地方法的调用状态；

方法区：当虚拟机装载一个class文件时，其会将class的类型信息（类信息、常量、静态变量等）放入方法区，该内存区域被所有线程共享；

堆：堆是被线程共享的一块内存区域，用于存放对象实例，即动态申请的资源，实例创建时，在栈中分配引用，并且将该引用指向堆中的实例。
Java中的垃圾回收机制负责回收堆内存中没有被使用的对象，即这些对象已经没有引用了。

14. 什么是虚拟内存？什么是共享内存？

答：虚拟内存是计算机系统内存管理中的一种技术，它使得进程认为其拥有连续可用的内存，即完全连续的地址空间。而实际上，其在物理上通常被分割为多个内存碎片，有部分暂时存储于外部磁盘上，在需要进行数据交换。这种虚拟技术同时也从逻辑上对内存进行了扩充，通过请求调入与置换算法实现对内存地址的扩容。其两个主要的优点是：

- ① 虚拟内存地址空间连续，没有碎片；
- ② 虚拟内存的最大空间是CPU的最大寻址空间，而不受内存大小限制。

共享内存是最快度的进程间通信机制，操作系统在几个进程的地址空间上映射一段内存，然后这几个进程可以在不需要调用系统函数的情况下在那段内存上进行读/写操作。

15. 有哪些页面置换算法？

答：① 最佳置换算法（OPT）：每次选择淘汰的页面是以后最长时间不再被访问的页面。这种算法是最理想的算法，目前无法实现，但可以作为衡量一个置换算法好坏的标杆；

② 先进先出置换算法（FIFO）：每次选择淘汰最先进入内存的页面，这种置换算法保证了公平性，但是会出现belady现象，即当内存空间增大时，页命中率下降；

③ 最近最久未使用算法（LRU）：每次淘汰的页面是最近最久未被使用的页面，该算法利用了栈的特性，因此不会出现belady现象；

④ 时钟置换算法（Clock）：为每个页面设置一个访问位，页面被访问时，将其访问位设置为1，每次系统选择淘汰访问位为0的页面。每次进行淘汰时，系统循环地检查每个页面访问位，若为1，则设置为0；若为0，则淘汰页面。Clock算法给了页面留存内存的第二次机会；

⑤ 改进时钟算法：在访问位的基础上添加一个修改位，两个位组合一共有四种组合：00,01,10,11。扫描到第一个00，则将页面淘汰；若第一轮失败，则第二轮扫描第一个01淘汰，并且将10设置为00，将11设置为01；若前两轮失败，则重复前两轮。因为此时10与11已经被更新为00和01，因此一定可以找到用于替换的页面。

16. 有哪些磁盘调度算法？

答：① 先来先服务（FCFS）：这是一种公平调度算法，其根据请求访问磁盘的先后顺序进行调度；

② 最短寻道时间优先（SSTF）：该算法考虑当前磁头距离哪个被请求的磁道，则哪个磁道将被优先访问。但是该算法可能产生饥饿现象；

③ 扫描算法（SCAN）：该算法是先满足一个方向所有的请求，再满足反方向的所有请求，如此循环往复。该算法如果在一个方向上已经没有请求，则可以提前掉头，无需扫描到末端；

④ 循环扫描算法（CSCAN）：该算法在SCAN算法的基础上，规定磁头只能单向移动，即只满足同一个方向上的所有请求；

⑤ NStepSCAN：为了解决SSTF、SCAN、CSCAN等算法产生的磁臂黏连现象，该算法将磁盘请求队列分成若干个长度的为N的子队列，并且按照FCFS依次处理这些子队列。在处理子队列的请求时，利用SCAN算法。

17. 磁盘存储管理有哪些方法？

答：主要有三种方法，分别是连续分配、链接分配和索引分配。

连续分配：要求每一个文件分配一组相邻接的盘块，访问速度快，但是必须事先知道文件的大小；

链接分配：分为隐式链接和显示链接，隐式链接是指在每个目录项中都含有指向链接文件第一盘块和最后一个盘块的指针，每个盘块都有指向下一个盘块的指针，显示链接是指把用于链接文件各物理块的指针都存放在内存中的一张链接表中。它不支持随机访问，但是利于文件的动态增长；

索引分配：分为单级索引分配以及多级索引分配。它为每个文件分配一个索引表，把分配给该文件的所有盘号都记录在该索引块中。它不能支持高效的直接存取。

18. 文件系统中文件是如何组织的？

答：文件的逻辑组织结构通常分为有结构文件和无结构文件，又称记录文件和字符流文件。其中，有结构文件即在逻辑上被视为一组连续记录的集合的文件，分为定长记录文件和不定长记录文件两种；无结构文件则是内部不在划分记录，而是由一组相关信息组成的有序字符流。

文件的物理结构通常分为顺序文件、索引文件和索引顺序文件。顺序文件是指由一系列记录按照某种顺序排列所形成的文件，索引文件是指为变长记录建立一张索引表，为每个记录设置一个表项，从而加速对记录的检索。索引顺序文件是为一组记录的第一个记录建立索引表。

19. OS对IO设备的控制方法有什么？

答：① 使用轮询的可编程IO方式：在处理机向控制器发出一条IO指令时，启动输入输出设备的同时将其状态寄存器设置为busy，然后系统不断进行轮询测试，检测IO设备是否繁忙。这种方式将浪费大量的CPU时间在询问IO设备是否繁忙上；

② 使用中断的可编程IO方式：当某个进程需要启动某个IO设备时，由CPU向该设备发出一条IO命令，而后返回继续执行原来的任务，设备控制器即按照命令去控制IO设备，完成任务之后，设备控制器向CPU发出一条中断信号；

③ 直接存储器访问方式DMA：中断方式是以字为单位进行IO，效率低下。而DMA的数据传输基本单位是数据块，并且整块数据的传输是在DMA的控制下完成，减少了CPU的干预。所传输的数据是直接从设备到内存，不需要经过CPU，提高了CPU的使用效率。

20. 中断处理程序包括哪几个步骤？

答：

1. 测定是否有未响应的中断信号。若没有，继续执行下一条指令；若有，则停止原有进程的执行，准备转去执行中断处理程序，为把处理机转交给中断处理程序做准备
2. 保护被中断的CPU环境。将CPU寄存器内容压入中断栈。
3. 转入相应的设备处理程序。由处理机对各个中断源进行测试，以确定引起本次中断的IO设备，并向提供中断信号的设备发送确认信号；设备收到信号后停止发送中断信号。然后将中断处理程序的入口地址装入到程序计数器中，处理机运行时，便可自动转向中断处理程序。
4. 中断处理。对于不同的设备，有不同的中断处理程序。程序首先从设备控制器读取状态，判断中断是正常完成中断还是异常结束中断，根据不同的中断类型

执行不同的操作。

5. 恢复CPU现场并退出中断。退出中断后，是否会返到被中断的进程，取决于2个因素：
6. 本次中断是否采用了屏蔽（禁止）中断的方式，若是，则返回被中断的进程。
7. 采用的是中断嵌套方式，若没有优先级更高的中断请求IO，在中断完成后会回到被中断的进程；反之，系统将处理优先级更高的进程。