操作系统面试题汇总

1. 什么是进程与线程，有何区别？

1. 进程是资源分配和管理的基本单位，每个进程都有自己独立的虚拟地址空间，线程是 cpu 调度和执行的基本单位，
2. 一个进程可以包含多个线程，一个线程只能属于一个进程。同一进程中的线程共享进程所拥有的全部资源。
3. 进程之间切换开销较大，线程之间切换开销较小。

cpu 运行到进程 A，先获取 A 运行的上下文，之后运行进程 A 中的 a,b,c 线程，然后保存 A 运行的上下文，这就是一个完整的进程的运行过程。

2. 进程有哪几种状态？

运行态：进程获得 cpu 时间片，运行完后会进入就绪态，遇到阻塞事件(如 IO 操作)会进入阻　塞态。

就绪态：进程已经获得了除 cpu 外的全部资源，只等 cpu 时间片就可以进入运行态。

阻塞态：进程由于某些阻塞事件(如 IO 操作)会进入阻塞态，阻塞事件处理完后会继续运行。

3. 操作系统中进程调度策略有哪几种？

1. 先来先服务
2. 短作业优先
3. 优先级调度
4. 高响应比优先(等待时间+运行时间/运行时间=响应比)
5. 时间片轮转
6. 多级反馈队列(设置多个队列，优先级高的时间片长度短，新来的作业先放到第一队列的末尾，如果该时间片内没执行完，则放到第二队列的末尾，以此类推)。

4. 进程间的通信方式有哪些？

管道、信号、信号量、共享内存、socket、消息队列。

管道：管道分为两种，匿名管道和有名管道。匿名管道用于有亲缘关系的进程之间，有名管道可以用于任意两个进程之间，管道是半双工的，常见的 bash 命令中的“|”就是管道命令符，是将前一个命令的结果传递给后一个命令。

信号：信号是 linux 中进程间相互通信的一种机制，例如 ctrl + c，kill -9 等都是不同的信号。

消息队列：存放在内核中的消息链表。

共享内存：多个进程可以同时读写同一块内存区域，是速度最快的进程间通信方式，需要注意进程间的同步与互斥问题。

信号量：通过 PV 操作来实现进程之间的同步。

套接字 socket：详细介绍见相关问题。

参考：https://www.jianshu.com/p/c1015f5ffa74

5. 孤儿进程，僵尸进程，守护进程是什么

父进程使用 fork()方法创建子进程，子进程退出后，父进程要调用 wait()方法来获得子进程的状态，除了 init 进程外，所有进程都是用 fork 进行创建的。

孤儿进程：父进程退出之后，子进程仍在运行，此时子进程会被 init 进程(进程号为 1)收养。

僵尸进程：子进程退出之后，父进程不调用 wait()方法，导致子进程的进程号依然被占用，产生僵尸进程，解决方法是杀死父进程，由 init 进程收养之后进行释放。

守护进程：守护进程是长期运行在后台的一类进程，它独立于终端存在并且周期性的执行某些功能，其父进程是 init 进程。

什么是写时复制：

调用 fork 系统调用创建子进程时，并不会把父进程所有占用的内存页复制一份，而是与父进程共用相同的内存页，而当子进程或者父进程对内存页进行修改时才会进行复制——

这就是著名的 写时复制 机制。

6. 说说进程间的同步方式

进程之间的独立性很高，进程之间的同步方式等同于进程之间的通信方式，参考前面的回答。

7. 什么是临界区？如何解决冲突？

临界资源是指某个时间同时只允许一个进程访问的资源，硬件资源如 I/O 设备，软件资源如共享变量，访问该资源的程序称为临界区，解决冲突的方式是忙则等待、有限等待，具体的加锁机制类似分布式锁。

8. 什么是死锁，如何避免

死锁是指多个进程之间因为资源冲突导致相互等待，从而无法无限期阻塞的问题，造成死锁的四个必要条件是：

1. 互斥：临界资源只能由一个进程使用
2. 不可剥夺：进程在使用完资源前不能强行剥夺
3. 占用并等待：一个进程在使用完资源之前会一直占用
4. 环路等待：几个进程对资源的等待形成一个环路

死锁避免就是要破坏以上几个条件，比如允许剥夺资源，主动释放资源、有序分配资源打破环路等待。

9. 说说线程间的同步机制

由于多个线程之间共享同一进程的数据，所以容易引发数据错误的现象。所以，我们需要一些方法去避免多个线程同时去操作共享资源（ 如多个线程都可以操作的全局变量）时产生的错误。常见的方法有(其实本质上都是锁的机制)：

1. 锁机制：互斥锁、读写锁(根据实际的场景确定)等。
2. 信号量：PV 操作，可以控制同一时间访问临界区的线程数。

11. 了解什么是协程吗

协程是一种用户态的轻量级线程，特点是：

1. 因为不需要进行用户态与内核态的翻转，所以协程之间切换的效率极高
2. 协程本质上是单线程的，无需线程上下文切换的开销
3. 线程和协程主要用在 IO 密集型的任务中，cpu 密集型的任务主要是用多进程进行并行处理。

12. 中断和轮询的特点

中断和轮询都是处理 I/O 请求的方式，中断是指 CPU 在正常运行程序的过程中，由于某些外部的事件导致 CPU 中断当前正在运行的程序，转而去处理该事件的事件的过程。轮询是指定时对所有 I/O 设备进行一次轮询，有相应请求的则进行处理，但这样比较低效。

13. 什么是虚拟内存

操作系统为每个进程分配了一个独立的虚拟地址空间，让每个进程感觉好像独占内存一样，实际上是利用页面置换的方式，只有使用时才将对应的页装入物理内存，这种虚拟内存的方法可以更加高效的管理内存，同时将磁盘空间也纳入了内存的管理的范畴，大大扩展了能够使用的空间，实际上是一种以时间换空间的策略。

14. 说说段页式管理

页是物理单位，一般来说一页的大小是 4KB，物理内存和虚拟内存都按页进行划分，并且通过页表进行映射。如果要访问的页不在内存中，则产生缺页中断，并且使用页面置换算法进行置换。

段是逻辑单位，段的大小不固定，通过段表进行物理内存和虚拟内存中段的映射，

分页管理的优点是内存利用率高，缺点是无法按照逻辑进行信息的管理。

分段管理的优点是可以按照逻辑进行信息的管理，缺点是段间的外部碎片大，内存利用率低。

段页式管理则是集合了两者的优势，首先按逻辑进行分段，然后对每段进行分页。这样一个完整的寻址过程就是：段表查找段号 => 该段的页表查找页号 => 通过页内偏移地

址找到最终的物理地址。

15. 为什么要使用多级页表

虚拟地址空间大小为 4GB，每个页表的大小为 4KB，那么页表项就需要有 1M 项，如果这么大页表连续存储并且常驻内存，会很大程度上降低内存使用率，因此使用多级页表的好处在于：

1. 使得页表可以离散存储，而不需要一块连续的内存
2. 页表本身不需要常驻内存，暂时使用不到的页表可以通过页面置换算法被置换到磁盘上。

15. 有哪些页面置换算法

页面置换是虚拟内存中背后的技术支撑，当程序运行所需的页面不在物理内存当中时，就需要进行页面置换，将磁盘中的页面置换到内存中，页面置换算法旨在降低缺页中断的概率。

1. FIFO：先进先出置换算法，置换最早换入的页面
2. LRU：最近未使用置换算法，每次使用到某页时就更新该页的使用时间，当需要置换时，选择最久未使用的页面进行置换。
3. LFU：最不常用置换算法，记录页的访问次数，每次置换时选择访问次数最少的页进行置换。

16. 局部性原理是什么

时间局部性：如果一个内存单元被引用，那么在未来可能会被多次引用。

空间局部性：如果一个内存单元被引用，那么它附近的内存单元很可能被引用。

举例就是对一个数组进行 for 循环求和，同时包含了时间局部性和空间局部性。

17. 什么是缓冲区溢出？

缓冲区溢出是指某些固定大小的缓冲区被赋予了超出其长度的内容，导致程序崩溃或者跳转到一段恶意代码上。一般指的 c 语言，java 中一般不会出现。

18. 什么是 IO 多路复用

IO 多路复用就是使用一个线程同时处理多个 IO 连接的技术。最早的计算机并发处理中，每新建一个 IO 连接，就要新建一个线程用于处理，当并发量过大时这种方案显然是不可接受的，于是有了 IO 多路复用，使用单线程同时处理多个连接，又分为以 select 和 poll 为代表的轮询方式以及 epoll 为代表的事件驱动方式。

19. select，poll 和 epoll 的区别是什么

Linux 中的 select、poll、epoll 都是 IO 多路复用技术，用于并发处理，其中 select 和 poll 是采取轮询的方式，能处理的并发量较少，epoll 是采取事件驱动的方式，在高并发的场景下依然有非常好的性能。

select 作为最先出现的 IO 多路复用技术，主要缺点是：

1. 每个进程能够监听的文件描述符有限，一般是 1024 个，那么高并发的场景下需要开启大量进程，会消耗大量资源。
2. select 采取轮询的方式判断是否有事件发生，效率比较低。
3. 每次需要复制数组到内核态在进行轮询，开销巨大
4. select 采取水平触发的机制(只要有未处理完的数据就会一直通知，而不是事件发生变化时再通知，类似于电路信号中的触发)，系统中会有大量不关心的就绪文件描述符

poll 与 select 非常相似，只是使用链表进行存储，没有了单进程最大连接数的限制。

epoll 的核心是三个 api(create，ctl，wait)及两个数据结构(红黑树，链表)。epoll create 时 会在系统内核新建一棵红黑树用于存放要监听的文件描述符，同时会新建一个链表用于存放所有就绪的文件描述符；epoll ctl 时会向红黑树插入一个描述符节点，同时注册一个回调函数用于当事件发生时向就绪链表中加入节点。epoll wait 时会查看就绪描述符链表，链表不为空则进行处理。

epoll 的优点在于：

1. 没有监听文件描述符数量的限制。
2. epoll 采取事件驱动机制，当有事件发生时会主动的通过回调函数向就绪链表中插入节点，而不是轮询的方式，因此效率高。

3. epoll 在内核中创建了相关的数据结构，避免了内核态与用户态之间的大量拷贝，开销很小。
4. epoll 采取边缘触发的方式，没有不关心的就绪文件描述符。

参考：https://zhuanlan.zhihu.com/p/159135478

个人理解：epoll 是异步非阻塞，select / poll 是同步非阻塞。

首先一个线程处理多个请求，本身就是非阻塞的；select / poll 主动进行轮询，是同步的，epoll 是事件驱动，从概念上讲是异步的。

20. 用户态和内核态的区别是什么

用户态和内核态最主要的区别是特权级不同，用来保障系统的安全，用户态是受限访问资源，内核态是不受限访问资源。

用户态切换到内核态的方式主要有三种，分别是：

1. 系统调用：这是用户主动申请切换到内核态的一种方式，操作系统专门开放了 ine 80h 中断用来进行系统调用。
2. 异常：当程序发生了异常时会切换到内核态进行处理，例如缺页异常。
3. 外围设备的中断：当外设发出 IO 请求时，会向 CPU 发出中断信号，此时需要切换到内核态进行请求的处理。

21. Linux 中常用到的命令

1. ls: -a 是可以展示出隐藏文件，-l 是可以展示出详细信息(比如权限、文件大小)
2. mkdir：创建目录，mkdir -p 可以创建出多个目录
3. rm：删除命令，-i 是逐个询问，-r 是删除文件夹，-f 是不询问强制删除
4. mv：移动文件或者重命名，第二个参数是文件名则重命名，是目录则移动到该目录下
5. cat：显示整个文件内容，创建文件、合并文件
6. more：类似 cat 展示文件的内容，但是可以按空格键逐页阅读
7. less：类似 more，但是 more 不能后退，less 可以

8. head：显示文件开头的内容
9. tail：显示文件末尾的内容
10. which：在$PATH 路径下查找可执行文件
11. whereis：在$PATH 路径下查找多种文件
12. locate：超快速查找任意文件，根据系统的索引数据库进行查找
13. find：直接搜索整个文件目录，功能强大但是速度慢
14. chmod：改变 linux 文件的访问权限，权限有三种：r(读)、w(写)、x(执行)
15. tar：压缩与解压
16. ln：为文件在另一个位置生成链接，软链接不占用空间，类似于快捷方式，硬链接会生成一个大小一样的文件，无论更改哪一个，另一个都会同步变化。
17. date：显示系统的时间与日期
18. grep：强大的文本搜索命令，用于在文件内容中搜索指定的字符串，-r 是递归查找，-l 是只显示文件名
19. wc：统计功能，统计文件中的行数、单词数、字节数
20. ps：查看当前进程的运行状态
21. top：查看每个进程的详细信息，包括内存占用，cpu 使用等。同时可以查看目前总的 cpu 使用率和内存占用率。
22. free：可以查看详细的内存使用情况
23. df：查看磁盘整体的使用情况
24. du：查看目录下每个文件的大小
25. kill：终止进程的命令，可以有多种信号，其中只有-9 能够无条件终止进程。

    kill -9 杀不死进程的原因：该进程为僵尸进程
26. netstat：查看网络连接状态，包括端口状态等
27. ifconfig：查看网络设备的状态，比如 ip 地址等
28. iostat：查看 IO 状态
29. vm_stat：虚拟内存使用状态

30. crontab：定时任务

31. wget：用于下载文件

32. curl：用于发送 http 请求

33. jobs -l：查看后台任务

34. alias：取别名

35. su：切换用户

36. awk：处理文本文件，逐行读取数据，常用于日志分析等。

参考：https://www.runoob.com/w3cnote/linux-common-command-2.html

参考文章：

1. https://juejin.cn/post/6844903639207641096
2. https://my.oschina.net/u/3827987/blog/3217890
3. https://blog.nowcoder.net/n/49211c67aaaa49eb8842f7e979c79498
4. https://zhuanlan.zhihu.com/p/23755202