853769620

1、面试题

进程间是如何通信的？线程间又如何切换呢？

2、面试官心里分析

这个问题不是高频基础问题，但是确实可能有人会问，因为怎么说呢，计算机基础，就这点儿东西，网络、cpu、磁盘、内存、进程，所以可能有人会看看你的基础知识咋样，所以问问你这个问题。

3、面试题剖析

进程间的通信有很多种方式，比如说：管道（pipe）、命名管道（fifo）、消息队列，共享内存（System V）

（1）管道（pipe）

unix操作系统里面，有一个fork操作，可以创建进程的子进程，或者说是复制一个进程完全一样的子进程，共享代码空间，但是各自有独立的数据空间，不过子进程的数据空间是拷贝父进程的数据空间的。

管道机制要求的是两个进程之间是有血缘关系的，就比如fork出来的父子进程。

linux操作系统里，管道用来缓存要在进程间传输的数据，管道是一个固定大小的缓冲区，是4kb。管道中的数据一旦被读取出来，就不在管道里了。但是如果管道满了，那么写管道的操作就阻塞了，直到别人读了管道的数据；反之如果管道是空的，那么读操作就阻塞了。就这个意思。管道一边连着一个进程的输出，一边连着一个进程的输入，然后就一个进程写数据，另外一个进程读数据，两个进程都没了，管道也就没了。管道是半双工的，就是数据只能流向一个方向，比如说你架设一个管道，只能一个进程写，另外一个进程读。

linux里面对管道的实现，是用了两个文件，指向了一个VFS（虚拟文件系统）的索引节点inode，然后VFS索引节点指向一个物理页面，接着一个进程通过自己关联的那个文件写数据，另外一个进程通过自己关联的那个文件读数据。

（2）命名管道（fifo）

管道的通信，要求必须是父子关系的进程间通信，就受到了限制，所以可以用命名管理来解决这个问题。

之前的管道，是没有名字的，所以必须是有父子关系的进程才能使用。但是这个命名管道是有名字的。这个命名管道，相当于是一个有名字的文件，是有路径的，所以没有血缘关系的进程多可以通过这个命名管道来通信，名字在文件系统上，数据在内存里。其他的跟管道一样，一个进程写，一个进程读，也是半双工的，数据只能单向流动。

（3）消息队列

linux的消息队列可以认为是个链表结构，linux内核有一个msgque链表，这个链表里每个指针指向一个msgid\_ds结构，这个结构就描述了一个消息队列。然后进程之间就通过这个消息队列通信就可以，一样是写入数据和消费数据。消息队列的好处就是对每个消息可以指定类型，消费的时候就消费指定类型的消息就行了，功能更多一些。这种方式其实用的不多的。

（4）共享内存

一块物理内存被映射到两个进程的进程地址空间，所以进程之间互相都可以立即看到对方在共享内存里做出的修改，但是因为是共享内存，所以需要锁来保证同步。这个说对了很复杂，我在这里就不多说了，我觉得如果被人问到这个问题，短期内突击的话，回答到这个程度就行了，就是知道有哪些方式。如果你要深入理解各种机制，那是要好好学习linux的各种东西了。

（5）线程间如何切换

一个进程的多个线程间切换的时候就涉及到了上下文切换，这个东西说复杂了就很复杂，但是简单来说，就是有一个时间片算法，cpu给每个线程一个时间片来执行，时间片结束之后，就保存这个线程的状态，然后切换到下一个线程去执行，这就是所谓多线程并发执行的原理，就是多个线程来回来去切换，每个线程就一个时间片里执行。太复杂的我也不讲了，大家就记住一个线程上下文切换指的是什么就行了。