**系统网络50问**

1. **标准文件IO与文件IO的区别？**

**答：**标准ＩＯ：标准I/O是ANSI C建立的一个标准I/O模型，是一个标准函数包和stdio.h头文件中的定义，拥有必定的可移植性。标准IO库处理很多细节。例如缓存分配，以优化长度执行IO等。标准的IO供给了三种类型的缓存:全缓存，行缓存不带缓存。

 文件ＩＯ：文件ＩＯ称之为不带缓存的IO（unbuffered I/O)。不带缓存指的是每一个read，write都调用内核中的一个系统调用。也就是一般所说的低级I/O——操作系统供给的基本IO服务，与os绑定，特定于linix或unix平台。

**区别：** 首先：两者一个明显的不同点在于，标准I/O默认采取了缓冲机制，比如调用fopen函数，不仅打开一个文件，而且建立了一个缓冲区（读写模式下将建立两个缓冲区），还创建了一个包含文件和缓冲区相干数据的数据结构。低级I/O一般没有采取缓冲，须要自己创建缓冲区，不过其实在linix或unix系统中，都是有应用称为内核缓冲的技术用于提高效率，读写调用是在内核缓冲区和进程缓冲区之间停止的数据复制。

其次从操作的设备上来区分，文件I/O重要针对文件操作，读写硬盘等，它操作的是文件描述符，标准I/O针对的是控制台，打印输出到屏幕等，它操作的是字符流。对于不同设备得特性不一样，必须有不同api访问才最高效。

**2、设备文件类型有哪些？**

**答：**Linux系统的设备文件分为三类:块设备文件，字符设备文件和网络设备文件。

块设备文件通常指一些需要以块的方式写入的设备，如IDE硬盘、光驱等。

字符型设备文件通常指可以直接读写，没有缓冲区的设备，如并口，虚拟控制台等

网络设备文件通常是指网络设备访问的BSDsocket接口，如网卡等。

**3、不缓冲、行缓冲和全缓冲的区别**

**答：全缓冲：知道缓冲区呗填满，才调用系统I/O函数。**对于读写操作来说，知道读入的内容的字节数等于缓冲区大小或者文件已经达到结尾，才进行实际的I/O操作，将外存文件内容读入缓冲区，对于写操作来说，直到缓冲区被填满，才进行实际的I/O操作，缓冲区内容写到外存文件中。磁盘文件通常是全缓冲的。

**行缓冲：直到遇到换行符’\n’,才调用系统I/O函数，**对于读操作来说，遇到换行符才进行I/O操作吧，讲所读内容读入缓冲区；对于写操作来说，遇到换行符才进行I/O操作，将缓冲区内容写到外存中。由于缓冲区的大小是有限的，所以当缓冲区被填满时，及时没有遇到换行符，也同样会进行实际的I/O操作。标准输入stdin和标准输出stdout默认都是行缓冲。

**不缓冲：没有缓冲区，数据会立即读入或者输出到外存文件个设备上。**标准出错stderr是无缓冲的，这样保证错误提示和输出能够及时反馈给用户，供用户排除错误。

**4、进程fork和vfork的区别**

**答：**fork创建一个进程时，子进程只是完全复制父进程的资源，这样得到的子进程独立于父进程，具有良好的并发性，但是二者之间的通讯需要通过专门的通讯机制，如：PIPE,FIFO,IPC机制等。通过fork创建的子进程系统开销很大，需要将每种资源(数据空间，堆和栈)都复制一个副本。这些系统开销并不是所有的情况下都是必须的。

vfork创建的子进程共享地址空间，也就是说子进程完全运行在父进程的地址空间上，子进程对虚拟地址空间任何数据的修改同样为父进程所见。但vfork创建子进程后，父进程会被阻塞知道子进程调用exec或exit。而fork的父子进程运行顺序是不定的，通过vfork可以减少不必要的开销。

**5、守护进程的编码规则？**

**答：**(1)调用umask将文件模式创建屏蔽字设置为0；

(2)创建子进程，同时父进程退出。

(3)调用setsid创建一个新的会话，并且担任会话组的组长。

(4)改变当前目录为根目录。

(5)重设文件权限掩码。

(6)关闭不在需要的文件描述符。

(7)打开/dev/null使其具有文件描述符0、1、2，禁止使用标准输入、输出、错误设备

(8)使用syslog(/dev/log)输出调试信息。

**6、进程通信中，同步信号与异步信号的差别？**

**答：**异步信号：进程通信在发送字符时，异步信号所发送的字符之间的时间间隔可以是任意的，不需要阻塞等待，

同步信号：双方必须要先建立同步，发送后等待接受，不可以任意收发。

**7、读锁和写锁的特点是什么？**

**答：**读锁与写锁互斥；读锁之间不互斥；写锁与写锁互斥。

一次只有一个线程可以占有写模式你的读写锁，但是可以有多个线程同时占有读模式的读写锁。

**8、3个PV原语经典模型是什么？**

**答：**生产者-消费者问题；读者-写者问题；哲学家就餐问题。

**9、进程的常见调度算法有哪些？**

**答：**先来先服务算法(FIFO)、时间轮转算法、优先级算法

**10、什么是进程，什么是线程？**

**答：**进程是计算机中的程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位，是操作系统结构的基础。它是系统资源管理的最小单位。

线程，有时别称为轻量级进程，是程序执行流的最小单元。线程是进程中的一个实体，是被系统独立调度和分配的基本单位，线程自己不拥有资源，只拥有一点儿在运行中必不可少的资源，他可以与同属一个进程的其他线程共享进程所拥有的全部资源。

**11、有了进程，为什么还要线程？**

**答：**进程有很多优点，它提供了多道编程，让我们感觉我们每个人都用有自己的CPU和其他资源，可以提高自己的利用率。但是进程只能在一个时间干一件事，在执行过程中如果阻塞，整个程序就会挂起。线程提高了进程的并发度，还可以有效的利用多处理器和多核计算机

**12、进程和线程的区别？**

**答：**(1)进程是系统进行资源分配的最基本单位，有独立的地址空间；线程是CPU调度的基本单位，没有单独的地址空间，有独立的栈、局部变量、寄存器等。

(2)创建进程的开销大，包括创建虚拟地址空间等需要大量系统资源；线程开销小，基本上只有一个内核对象和一个堆栈。

(3)一个进程无法直接访问另一个进程的资源；同一进程内的多个线程共享进程的资源。

(4)进程切换开销大，线程切换开销小；进程间通信开销大，线程间通信开销小。

(5)线程属于进程，不能独立执行。每个进程至少要有一个线程，成为主线程。

**13、用什么函数创建进程？**

**答：**fork(); vfork(); clone(); exec函数族。

**14、什么函数创建线程？**

**答：**pthread\_create()

**15、进程有返回值吗？可以返回几个？**

**答：**有，可以返回两个；例如fork()函数，调用一次缺返回两次；向父进程返回子进程的ID，向子进程中返回0。

**16、什么是僵尸进程？有什么危害？以及处理僵尸进程的方法。**

**答：**一个进程使用fork()创建子进程，如果子进程退出，而父进程并没有调用wait或者waitpid获取子进程的状态信息，那么子进程的进程描述符仍然保存在系统中。我们称之为僵尸进程。

危害：浪费系统资源，如果僵尸进程过多，那么就会有大量的进程号被僵尸进程占用，但系统所能使用的进程是有限。将会因为没有可用的进程号而导致系统不能产生新的进程。

处理方法：1.父进程等待及调用wait或waitpid；2.杀死父进程，父进程死后僵尸进程会变为孤儿进程，过继给init进程，init始终会负责清理僵尸进程，由它所产生的所有僵尸进程也跟着消失。

**17、什么是孤儿进程？**

**答：**一个父进程退出，而他的一个或多个子进程还在运行，那么这些子进程将会成为孤儿进程。孤儿进程将被init进程收养，并由init进程对他们完成状态收集工作。

**18、什么是后台进程？**

**答：**Linux后台进程也叫守护进程，是运行在后台的一种特殊进程。他独立于控制终端并且周期性的执行某种任务或等待处理某些发生的事件。

**19、fork()一个子进程后，父进程的全局变量能不能使用？**

**答：**fork后子进程将会拥有父进程的几乎一切资源，父子进程都各自有自己的全局变量。不能通用，不同于线程。对于线程，各个线程共享全局变量。

**20、什么是脱离线程？**

**答：**不向主线程返回信息，不需要主线程等待。

**21、线程取消有几种模式？**

**答：**立即取消和延迟取消。立即取消是调用pthread\_cancel的时候，不管线程当前在干什么，马上被结束掉。延迟取消是调用pthread\_cancel后，线程运行到一个取消点函数的时候 才会结束。

**22、什么是线程取消点？**

**答：**根据POSIX标准，pthread\_jion(), pthread\_testcancel(), pthread\_cond\_wait(), pthread\_cond\_timewait(),sem\_wait(),sigwait()函数以及read(),write()等会引起阻塞的系统调用都是线程取消点。

**23、线程间的同步方式？**

**答：**临界区、互斥量、信号量、事件。

**24、进程间通信的方法有哪些？各自有什么优缺点？那一种方法效率最高？**

**答：无名管道（亲源进程之间）：**管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。

**有名管道（任意进程之间）：**也是半双工通信方式，但是允许在没有亲缘关系的进程之间使用，管道是先进先出的通信方式。

**共享内存（效率最高）：**共享内存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，他是针对其他进程间通信方式运行效率低而专门设计的。

**消息队列（具备同步的效果）：**消息队列是有消息的链表，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。

**socket（不同主机之间的进程通信）：**用于不同机器间的进程通信。

**信号量（同步机制）：**是一个计数器，可以用来控制多个进程对共享资源的访问。它常作为一种机制锁，防止某进程在防伪共享资源时，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。

**内存映射：**

共享内存效率最高

**25、线程之间通过什么交换数据**

**答：**全局变量；Message消息机制；CEvent对象

**26、线程条件变量是起什么作用？**

**答：**条件变量是利用线程间共享的全局变量进行同步的一种机制，主要包括两个动作:一个线程等待某个条件为真，而将自己挂起；另一个线程使得条件成立，并通知等待的线程继续。为了防止竞争，条件变量的使用总是和一个互斥锁结合在一起。当条件变量没有被链接到特定的断言上去，当一个条件变量与多个断言相关联的时候，线程被唤醒后就必须重新测试它所要的断言。其次条件变量可以让线程睡眠特定的时间。

1. **启动Linux Shell时，默认打开哪几个文件描述符，他们的值分别是多少？**

**答：**stdin;stdout;stderr; 0,1,2

1. **内核主要分为哪几个子系统？**

**答：**进程管理系统、内存管理系统、I/O管理系统和文件管理系统等四个子系统。

1. **操作系统中进程调度策略有哪几种？**

**答：**FCFS(先来先服务)、优先级、时间轮转片、多级反馈。

1. **进程具有那些基本状态？**

**答：**新建、就绪、运行、阻塞、退出

1. **TCP和UDP的区别？**

**答：**(1)TCP面向连接(三次握手机制)、通信前需建立连接；UDP面向无连接，通信前不需要建立连接；

(2)TCP保障可靠传输(按序、无差错、不丢失、不重复)；UDP不保障可靠传输，使用最大努力交付；

(3)TCP面向字节流的传输，UDP面向数据报的传输。

**32、TCP建立连接的时候三次握手，断开连接时的四次握手的具体过程？**

**答：建立连接的三次握手-------**

第一次握手是客户端connect连接到server，server accept client的请求之后，向client端发送一个消息，相当于说我都准备好了，你连接上我了，这是第二次握手，第三次握手就是client向server发送的，就是对第二次握手消息的确认，之后server和client就开始通讯了。

**断开连接的四次握手-----**

断开连接的一端发送close请求是第一次握手，另外一端接收到断开连接的请求之后需要对close进行确认，发送一个消息，这是第二次握手，发送了确认消息之后还要向对端发送close消息，要关闭对对端的连接，这是第三次握手，而在最初发送断开连接的一端接收到消息之后，进入到一个很重要的状态time\_wait状态，最后一次握手是最初发送断开连接的一端接收到消息之后对消息的确认。

1. **connect方法会阻塞，请问有什么方法可以避免其长时间阻塞？**

**答：**最常用且最有效的是加定时器；也可以采用非阻塞模式。

或者考虑采用异步传输机制，同步传输与异步传输的主要区别在于同步传输中，如果recvfrom后会一直阻塞到运行，从而导致调用线程暂停运行；异步传输机制则不然，会立即返回。

1. **网络编程中设计并发服务器，使用多进程与多线程，请问有何区别？**

**答案一：**进程是父进程的复制品。子进程获得父进程数据空间、堆和栈的复制品。

线程相对于进程而言，线程是一个更加接近于执行体的概念，他可以与同进程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。

两者都可以提高程序的并发度，提高程序运行效率和响应时间。

线程和进程使用上各有优缺点：线程执行开销小，但不利于资源管理和保护；而进程正相反。同时，线程适合于SMP机器上运行，而进程则可以跨机器迁移。

**答案二：**根本区别就一点：用多进程每个进程有自己的地址空间，线程则共享地址空间，所有其他区别都是 由此而来的：

速度：线程产生的速度快，线程间的通讯快，切换快等， 以为它们在同一个地址空间内。

资源利用率：线程的资源利用率比较好也是因为它们在同一个地址空间内。

同步问题：线程使用公共变量、内存是需要使用同步机制还是因为它们在同一个地址空间内。

1. **TCP为什么不是两次连接而是三次握手？**

**答：**如果A与B两个进程通信，如果仅是两次连接，可能出现的一种情况是：A发送完请求报文以后，由于网络情况不好，出现了网络拥塞，即B延时很长时间后收到报文，即此时A将此报文认定为失效的堆报文。B收到报文后，会向A发起连接。此时两次握手完毕，B会认为已经建立了连接可以通信，B会一直等到A发送的连接请求，而A对失效的报文回复自然不会处理。一次会陷入B忙等的僵局，造成资源的浪费。

1. **TCP的重发机制是怎么实现的？**

**答：**(1)滑动窗口机制，确认收发的边界，能让发送方知道已经发送了多少(已确认)、尚未确认的字节数，尚等待发送的字节数；让接收方知道(已经确认接收到的字节数)。

(2)选择重传，用于对传输出错的序列进行重传。

1. **UDP分多少种形式，各有什么特点？**

**答：**单播：一对一的通讯模式，服务器及时响应客户机的请求。

多播/组播：一对一组的通讯模式，需要相同数据流的客户端加入相同的组共享一条数据流，节省了服务器的负载。

广播：一对所有的通讯模式，所有主机都可以接收到所有信息，服务器流量负载极低。

1. **解释单体内核和微内核之间的区别？**

**答：**单体内核包含了所有功能：调度，文件系统，设备驱动程序，网络，存储管理等。

微内核只有部分功能，基本调度，进程通信，地址空间。

1. **网络I/O的五种模式？**

**答：**阻塞I/O，非阻塞I/O，信号驱动，I/O复用，异步I/O。

1. **TCP/IP五层模型？**

**答：**应用层，传输层，网络层，数据链路层，物理层。

1. **文件描述符和FILE\*的关系？**

**答：**文件描述符：在linux系统中打开文件就会获得文件描述符，是一个小整数。每个新城在PCB中保存着一份文件描述符表，文件描述符就是这个表的索引

文件指针：C语言的标准库使用文件指针作为文件的句柄，文件指针指向进程用户区的一种FILE类型的数据结构，FILE结构中包含一个文件描述符域和一个缓冲区，文件描述符是文件描述符表的一个索引

1. **TCP和UDP的用途？**

**答：**TCP一般用于文件传输，发送或者接受邮件，远程登录等等；

UDP一般用于即时通信，在线视频，网络语音通话等等。

**43、TCP四次分手中，主动关闭方最后为什么要等待2MSL之后才关闭连接？**

**答：**这是因为双方都同意关闭连接了，而且握手的四个报文也都协调和发送完毕，按理可以直接回到CLOSED状态；但是因为如果网络是不可靠的，无法保证最后发送的ACK报文会一定被对方接受到，因此对方处于 ＬＡＳＴ＿ＡＣＫ状态下的ＳＯＣＫＥＴ可能会因为超时未收到ＡＣＫ报文而重新发送ＦＩＮ报文，所以这个ＴＩＭＥ＿ＷＡＩＴ状态的作用就是用来重发可能丢失的ＡＣＫ报文。

1. **网络中，如果客户端突然关掉或者重启，服务器怎么样才能立刻知道？**

**答：**若客户端突然掉线或者重启，服务器端会收到复位信号，每一种tcp/ip的实现不一样，控制机制也不一样。

1. **TTL是什么？有什么用处？通常哪些工具会用到它？**

**答：**TTL是Time To Live一般是hub count每经过一个路由就会被减去一，如果他变成0，包会被丢掉。他的主要目的是防止包在有回路的网络上死转，浪费网络资源。ping和traceroute用到它。

1. **什么是IP协议？在哪个层面上？主要有什么作用？**

**答：**IP协议是网络层的协议，他是为了实现相互连接的计算机进行通信设计的协议，它实现了自动路由功能，及自动寻径功能。

1. **描述RARP协议。**

**答：**RARP是逆地址解析协议，作用是完成硬件地址到IP地址的映射，主要用于无盘工作站，因为给无盘工作站配置的IP地址不能保存。

工作流程：在网络中配置一台RARP服务器，里面保存着IP地址和MAC地址的映射关系，当无盘工作站启动后，就封装一个RARP数据包，里面有其MAC地址，然后广播到网络上去，当服务器收到请求包后，就查找对应的MAC地址的IP地址装入响应报文中发回给请求者。以为需要广播请求报文因此RARP只能作用于具有广播能力的网络。

1. **TCP/IP五层模型各层的作用。**

**答：**应用层：应用程序间沟通的层，如简单的电子邮件传输(SMTP),文件传输协议(FTP),网络远程访问协议等。

传输层：在此层中，它提供了节点间的数据传送服务，如传输控制协议(TCP),用户数据报协议(UDP).TCP和UDP给数据包加入传输数据并把它传输到下一层中，这一层负责传送数据等，并且确定数据已被送达并接收。

网络层：是TCP/IP协议组中非常关键的一层，主要定义了IP地址格式，从而能够使得不同应用类型的数据在Internet上畅通的传输，IP协议就是一个网络层协议。

数据链路层：这是TCP/IP软件的最底层，负责就收IP数据包并通过网络发送，或者从网络上接受物理帧，抽出IP数据报，交给IP层。

1. **交换机和路由器分别实现的原理分别是什么？**

**答：**交换机用于局域网，利用主机的MAC地址进行数据的传输，而不需关心IP数据包中的IP地址，它工作于数据链路层。路由器识别网络是通过IP数据包中的IP地址的网络号进行的，所以为了保证数据包路由的正确性，每个网络都必须有一个唯一的网络号，路由通过IP数据包的IP地址进行路由的(将数据包递交给哪一个下一跳路由器)。路由器工作于网络层，由于设备的发展，现在很多设备具有交换又具有路由功能，两者的界限越来越模糊。

1. **IP地址的分类。**

**答：** 网络号 网络范围 主机号

A类： 8bit 0------127 24bit

B类： 16bit 128.0--191.255 16bit

C类： 24bit 192.0.0---223.255.255 8bit

D类： 前四位固定为1110，后面为多播地址，所以D类为多播地址

E类： 前五位固定为11110，后面保留为今后所用

1. **程序什么时候应该使用线程，什么时候单线程效率高？**

**答：**(1)耗时的操作使用线程，提高应用程序响应。

(2)并行操作时使用线程，如C/S架构的服务器端并发线程响应用户的请求

(3)多CPU系统中，使用线程提高CPU利用率

(4)改善程序结构。一个既长又发杂的进程可以考虑分为多个线程，成为几个独立或半独立的运行部分，这样的程序会利于理解和修改。

其他情况都使用单线程。

1. **使用线程是如何防止出现大的波峰？**

**答：**意思是如何防止同时产生大量的线程。方法是使用线程池，线程池具有可以同时提高调度效率和限制资源使用的好处，线程池中的线程达到最大数时，其他线程就会排队等候。

1. **什么是线程池？**

**答：**线程池是指在初始化一个多线程应用程序中创建一个线程集合，然后在需要执行新的任务时，重用这些线程而不是新建一个线程。(为一个线程预分配一个集合或者一个池来已被未来之需以及能够在一个应用程序中重用的技术，称作线程池)

1. **线程池的作用以及为什么要用线程池？**

**答：**作用是限制系统中执行线程的数量。

根据系统的环境情况，可以自动或手动设置线程数量，达到运行的最佳效果；线程少了 浪费系统资源，多了造成系统资源拥挤，效率不高。用线程池控制线程的数量，其他线程排队等候。一个任务执行完毕，再从队列中亲缘取最前面的任务开始执行。若是队列里没有等待进程，线程池的这一资源处于等待。当一个新任务需要运行时，如果线程池中有等待的工作的线程，就可以开始运行了；否则进入等待队列。

为什么要用线程池：

(1)减少了创建线程和销毁线程的次数，每个工作线程都可以被重复利用，可以执行多个任务。

(2)可以根据系统的承受能力，调整线程池中工作的线程的数目，防止因为消耗过多的内存，导致服务器死机。