3. 当浏览器输入一个网站，接下来会发生什么？  
DNS解析：将域名解析成IP地址  
TCP连接：TCP三次握手  
服务器处理请求并返回HTTP报文  
浏览器解析渲染页面  
断开连接：TCP四次挥手

4. Linux epoll的LT（level triggered）与ET（edge-triggered）工作方式的区别？   
       LT:水平触发，效率会低于ET触发，尤其在大并发，大流量的情况下。但是LT对代码编写要求比较低，不容易出现问题。LT模式服务编写上的表现是：只要有数据没有被获取，内核就不断通知你，因此不用担心事件丢失的情况。  
　　ET:边缘触发，效率非常高，在并发，大流量的情况下，会比LT少很多epoll的系统调用，因此效率高。但是对编程要求高，需要细致的处理每个请求，否则容易发生丢失事件的情况。  
　　从本质上讲：与LT相比，ET模型是通过减少系统调用来达到提高并行效率的。

8. socket在什么情况下可读?

1. 接收缓冲区有数据，一定可读 2. 对方正常关闭socket，也是可读 3. 对于侦听socket，有新连接到达也可读

4.socket有错误发生，且pending～～～

9. 什么是对称加密与非对称加密

对称密钥加密是指加密和解密使用同一个密钥的方式，这种方式存在的最大问题就是密钥发送问题，即如何安全地将密钥发给对方；

而非对称加密是指使用一对非对称密钥，即公钥和私钥，公钥可以随意发布，但私钥只有自己知道。发送密文的一方使用对方的公钥进行加密处理，对方接收到加密信息后，使用自己的私钥进行解密。

由于非对称加密的方式不需要发送用来解密的私钥，所以可以保证安全性；但是和对称加密比起来，非常的慢

11路由表示做什么用的？在linux环境中怎么来配置一条默认路由？  
答:  
简:路由表是用来决定如何将包从一个子网传送到另一个子网的，换局话说就是用来决定从一个网卡接收到的包应该送的哪一张网卡上的。在Linux上可以用“route add default gw <默认路由器IP>”来配置一条默认路由。

12在网络中有两台主机A和B，并通过路由器和其他交换设备连接起来，已经确认物理连接正确无误，怎么来测试这两台机器是否连通？如果不通，怎么来判断故障点？怎么排除故障？  
答:测试这两台机器是否连通：从一台机器ping另一台机器  
     如果ping不通，用traceroute可以确定是哪个路由器不能连通，然后再找问题是在交换设备/hup/cable等。

13.TTL是什么？有什么用处，通常那些工具会用到它？（ping? traceroute? ifconfig? netstat?）  
答:  
简:TTL是Time To Live，一般是hup count，每经过一个路由就会被减去一，如果它变成0，包会被丢掉。它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。  
  
详:TTL是Time To Live，目前是hup count，当包每经过一个路由器它就会被减去一，如果它变成0，路由器就会把包丢掉。IP网络往往带有环(loop)，比如子网A和子网B有两个路由器相连，它就是一个loop。TTL的主要目的是防止包在有回路的网络上死转，因为包的TTL最终后变成0而使得此包从网上消失(此时往往路由器会送一个ICMP包回来，traceroute就是根据这个做的)。ping会送包出去，所以里面有它，但是ping不一定非要不可它。traceroute则是完全因为有它才能成的。ifconfig是用来配置网卡的，netstat -rn 是用来列路由表的，所以都用不着它

16 以下代码输出结果：

int main(void)  
{  
printf("call execl");   
sleep(1);   
execl("/bin/sh", "", NULL);   
printf("error!\n");  
}  
本题考标准IO缓冲，标准出错是不带缓缓冲的。

如若是涉及终端设备的其他流，则他们是行缓冲的；否则是全缓冲的。

printf是标准IO的一个，格式化打印到标准输出，在这里是行缓冲，那么没有遇到换行符也就是‘\n’或者没有强制flush, 则不会输出。

execl是创建新的可执行程序映像，一旦成功就不会返回了，只有在出错的情况会返回1.

所以以上的程序没有打印printf的内容，直接执行/bin/sh，输出为

$

若是代码改为以下：

int main(int argc, char \*argv[])

{

    printf("call execl\n");

    /\*fprintf(stderr, "%s", "call execl");\*/

    sleep(1);

    execl("/bin/sh", "", NULL);

    printf("error!\n");

    return 0;

}

则输出为：

call execl

$

若改为：

int main(int argc, char \*argv[])

{

    /\*printf("call execl\n");\*/

    fprintf(stderr, "%s", "call execl");       // 标准错误，不缓冲

    sleep(1);

    execl("/bin/sh", "", NULL);

    printf("error!\n");

    return 0;

}

则输出为：

call execl$

17**交换机与路由器有什么区别？**

①工作所处的OSI层次不一样，交换机工作在OSI第二层数据链路层，路由器工作在OSI第三层网络层

②寻址方式不同：交换机根据MAC地址寻址，路由器根据IP地址寻址

③转发速不同：交换机的转发速度快，路由器转发速度相对较慢。

# 18如何解决TIME\_WAIT过多的解决办法

在linux上可以这么配置：  
#让TIME\_WAIT状态可以重用，这样即使TIME\_WAIT占满了所有端口，也不会拒绝新的请求造成障碍  
echo "1" > /proc/sys/net/ipv4/tcp\_tw\_reuse  
#让TIME\_WAIT尽快回收，我也不知是多久，观察大概是一秒钟  
echo "1" > /proc/sys/net/ipv4/tcp\_tw\_recycle  
很多文档都会建议两个参数都配置上，但是我发现只用修改tcp\_tw\_recycle就可以解决问题的了，TIME\_WAIT重用TCP协议本身就是不建议打开的。

19对于使用tcp通信的两端，如果client已经退出，此时服务端继续send会出现什么问题？这个当然就要扯到SIGPIPE信号了？

20

NAT是什么？底层实现原理？

21半连接队列？全连接队列？

22

TCP参数调优

对于三次握手来说的74

开启SYN Cookies。当出现SYN等待队列溢出时，启用cookies来处理，可防范少量SYN攻击，默认为0，表示SYN Cookies关闭。

Net.ipv4.tcp\_syncookies=1

Cookies：简单来说cookies就是浏览器储存在用户电脑上的一小段文本文件。Cookies是纯文本格式，不包含任何可执行的代码。Cookies的本质是一个键值对，由键值key、value构成，键值对之间由一个分号和一个空格隔开，当浏览器访问web服务器的时候写入在客户端机器上，里面记录一些信息。一个web服务器会告知浏览器按照一定规范来储存这些信息，并在随后的请求中将这些信息发送至服务器，Web服务器就可以使用这些信息来识别不同的用户。大多数需要登陆的网站在用户验证成功之后都会设置一个cookie，只要这个cookie存在这个用户，他就可以自由浏览这个网站的任意界面。

SYN攻击原理

SYN攻击属于DOS攻击的一种，它利用TCP协议缺陷，通过发送大量的半连接请求，耗费CPU和内存资源。

SYN攻击除了能影响主机外，而可以危害路由器、防火墙等网络系统，事实上SYN攻击并不管目标是什么系统，只要这些系统打开TCP服务就可以实施。服务器接收到连接请求（syn=x），将此消息加入未连接队列，并发送ACK+SYN确认包给客户。当服务器未收到客户端的确认包时（第三次握手），重发请求包，一直到超时，才将此条目从未连接队列删除。配合IP欺骗，SYN攻击能达到很好的效果，通常，客户端在短时间内伪造大量不存在的IP地址，向服务器不断发送syn包，服务器回复确认包，并等待客户的确认，由于源地址IP是不存在的，服务器需要不断地重发直至超时，这些伪造地SYN包长时间占用未连接队列，正常的SYN请求被丢弃，目标系统运行缓慢，严重者引起网络阻塞甚至系统瘫痪。

系统默认的TIMEOUT时间

Net.ipv4.tcp\_fin\_timeout=5

当keepalive启用的时候，TCP发送keepalive消息的频度。缺省是2小时，改为20分钟。

Net.ipv4.tcp\_keepalive\_time=1200

表示用于向外连接的端口范围。缺省情况下很小：32768到61000改为10000到65000

Net.ipv4.ip\_local\_port\_range=10000 65000

SYN队列的长度，默认为1024，加大队列长度为8192，可以容纳更多等待连接的网络连接数

Net.ipv4.tcp\_max\_syn\_backlog=8192

系统保持TIME\_WAIT最大数量，如果超过这个数字，TIME\_WAIT将立刻被清除并打印警告信息。默认为180000，改为500.

Net.ipv4.tcp\_max\_tw\_buckets=5000

23如果你的系统接入大量的短连接，你认为哪里会出现性能瓶颈点

24mysql了解吗，查询慢怎么办

没有索引或者没有用到索引(这是查询慢最常见的问题，是程序设计的缺陷)

25   
so\_pcb表示socket控制块，其又指向一个结构体，该结构体包含了当前主机的ip地址(inp\_laddr)，当前主机进程的端口号(inp\_lport)，发送端主机的ip地址(inp\_faddr)，发送端主体进程的端口号(inp\_fport)。