

1 说明一下 tcp/ip 协议层次， 并介绍每层有哪些协议

应用层 (http ftp dns dhcp smtp ...)  
传输层 (tcp udp)  
网络层 (ip)  
网络接口层 (arp 以太网协议(有线网卡) 802.11(wifi) )

2 描述 socket 通信过程， 每个函数

tcp 通信 udp 通信

tcp 通信为例

服务器 socket bind listen accept recv/send close  
客户端 socket connect send/recv close

3 解释 tcp/ip 构造包的过程

在每一层加头部 (比如 经过 http 那么加 http 头, 经过 tcp 那么加 tcp 头....)

4 交换机和路由器的区别

路由器: 一台主机想要连接其他网络中的电脑, 那么需要经过路由器(网关)

交换机: 同一个网络中不同主机之间互相访问, 需要经过交换机

路由器: 寻址, 转发(依靠 IP 地址)

交换机: 过滤, 转发(依靠 Mac 地址)

5 端口号是干什么用的

端口号用来区分应用程序, 如果 A 程序绑定了 8888 端口号, B 程序绑定了 9999 端口号, 此时网络中来了一包数据, 系统发现是 8888 端口号, 就把数据给 A, 9999, 就给 B

http : 80

ftp : 21

DNS : 53

DHCP : 67, 68

6 有抓过包吗? (一定要具体了解具体抓它的问题) (小米)

用过: 学习 tcp/up 协议时用过 wireshark, 捕获数据包

共能: 1) 分层列出每个协议的报文, 可以对其进行分析

2) 可以根据协议过滤报文

7 腾讯实习生面试题

1) tcp 服务器创建时端口号冲突如何解决。

(1)bind 函数绑定时, 如果出错, 那么换一个端口号

(2)可以用 netstat 命令检查, 哪些应用程序使用我需要的端口号

netstat -anp | grep 8888

kill 杀掉进程

2) 说一下 tcp 服务器创建流程。

服务器 socket bind listen accept recv/send close

---

3) 然后共享屏幕敲代码写一个 tcp 服务器。

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <pthread.h>
void *recv_fun(void *p)
{
    int *q = p;
    int newfd = *q;    //int newfd = *(int *)p;
    char buf[100] = { 0 };
    while(1)
    {
        if(recv(newfd, buf, sizeof(buf), 0) <= 0)
            break;
        send(newfd, buf, sizeof(buf), 0);
        printf("recv is %s\n", buf);
    }
}
int main()
{
    pthread_t id;
    int fd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in myaddr;
    myaddr.sin_family = AF_INET;
    myaddr.sin_port = htons(9999);
    myaddr.sin_addr.s_addr = htonl(INADDR_ANY); //INADDR_ANY 得到当前计算机的 ip
地址
    int ret = bind(fd, (struct sockaddr *)&myaddr, sizeof(myaddr));
    listen(fd, 5);
    while(1)
    {
        int newfd = accept(fd, NULL, NULL); //等待客户端连接
        printf("client connect %d\n", newfd);
        pthread_create(&id, NULL, recv_fun, &newfd);
    }
    close(fd);
}
```

8 tcp 头多大, 里面有什么内容 (天融信)

tcp 头 20 个字节

16 位源端口号 16 位目的端口号

32 位序号 32 位确认序号

9 tcp 和 udp 的区别, 如果让你实现一个类似于 qq 的软件, 用 udp 还是 tcp

---

区别:

tcp 是一个安全可靠的传输协议(发送过程中发送方要保证接收方能收到)

1) tcp 通信前需要连接

2) 发送过程中, 接收方要回送应答信号, 如果发送方没收到应答信号, 重传

udp 不可靠的传输协议

1) 不管对方是否在线, 都发

2) 没收到, 也不重传

如果传输大量的数据(大文件、视频) 可以用 udp, 如果网络环境不好, 可以用 tcp

QQ 在视频聊天、文件传输 ... 时用 udp

在登录、注册时可以用 tcp

10 客户端如何检测服务器 是否 断开连接

recv 阻塞接收

方法 1: 通过 recv 函数的返回值, 如果返回值  $\leq 0$  说明对方断开了连接

方法 2: 可以采用心跳包机制

客户端和服务器每隔 1 秒钟进行握手操作(客户端发: 在不在 服务器回: 在)

11 用没用过线程池 (学过 用过)

tcp 并发服务器(多线程技术)

缺点: 1) 响应不及时 (每当客户端连接时, 要创建线程, 断开时, 要销毁线程, 耗时)

我们可以用线程池解决这个问题

1) 在系统启动时, 服务器创建先创建好多个线程, 线程先休眠

2) 一旦有任务需要执行, 马上唤醒线程, 执行相应任务, 然后继续休眠

线程池实现方法:

线程池 用 队列+链表方式进行管理

12 tcp 连接过程 (3 次握手), 断开过程(4 次端口)