



Socket 编程

TCP 阻塞方式

1551265 张伯阳



0.

cp 出一个 ifcfg-ens33:0 然后修改对应参数

```
● ● ● root@RHEL-zby:/etc/sysconfig/network-scripts (ssh)
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens33
UUID=fdd4a883-eba3-485d-8a8a-6a621e679285
DEVICE=ens33:0
ONBOOT=yes
IPADDR=192.168.3.231
PREFIX=24
IPv6_PRIVACY=no
~
~
~
~
~
~
"/etc/sysconfig/network-scripts/ifcfg-ens33:0" 18L, 330C
```

重启网络服务 再 ifconfig 查询 新 ip 已生成

```
[root@RHEL-zby ~]# systemctl restart network
[root@RHEL-zby ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.2.230 netmask 255.255.255.0 broadcast 192.168.2.255
              inet6 fe80::5afa:2209:89b3:6073 prefixlen 64 scopeid 0x20<link>
                ether 00:0c:29:05:21:df txqueuelen 1000 (Ethernet)
                  RX packets 140 bytes 17958 (17.5 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 106 bytes 22065 (21.5 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.3.231 netmask 255.255.255.0 broadcast 192.168.3.255
              ether 00:0c:29:05:21:df txqueuelen 1000 (Ethernet)
```

ping 另一个虚拟机 通

```
[root@RHEL-zby ~]# ping 192.168.2.231
PING 192.168.2.231 (192.168.2.231) 56(84) bytes of data.
64 bytes from 192.168.2.231: icmp_seq=1 ttl=64 time=0.543 ms
64 bytes from 192.168.2.231: icmp_seq=2 ttl=64 time=0.567 ms
64 bytes from 192.168.2.231: icmp_seq=3 ttl=64 time=0.522 ms
^C
--- 192.168.2.231 ping statistics ---
```

1.

建立目录不再赘述

2.

```
[root@RHEL-zby test]# ./client 192.168.2.231 4000
Hello
Hello
bye
bye
```

```
[root@RHEL-zby test]# ./server 4000
Hello
bye
```

若端口已被使用 则从报错可看出 是 bind 函数出错

```
(bind(server_sockfd,(struct sockaddr *)&server_sockaddr,sizeof(server_sockaddr))== -1)
```

```
[root@RHEL-zby test]# ./server 80
bind: Address already in use
```

IP 地址不正确 是 accept 函数出错 返回值为负数

```
conn = accept(server_sockfd, (struct sockaddr*)&client_addr, &length);
```

```
[root@RHEL-zby test]# ./client 192.168.2.233 4000
connect: No route to host
[root@RHEL-zby test]#
```

端口不正确 是 accept 函数出错 返回值为负数

```
accept(server_sockfd, (struct sockaddr*)&client_addr, &length);
```

```
[root@RHEL-zby test]# ./client 192.168.2.231 4001
connect: Connection refused
[root@RHEL-zby test]#
```

连接成功并给出提示

```
[root@RHEL-zby test]# ./server 4000
连接成功 !
[ ]  
X root@RHEL-zby:/home/homework/1551265-000107/test (ssh)
[root@RHEL-zby test]# ./client 192.168.2.231 4000
连接成功 !
[ ]
```

最开始 ctrl+c 或 kill -9 退出两端 对面都无法监测到 这里不再截图

原因是:connect 返回了 0 但对面没有对返回 0 进行处理只是单纯的阻塞在 recv

只需要在 recv 加判断 若其返回 0 将死循环 break 掉即可

修改代码为

```
cout<< "连接成功" << endl;
while(1)
{
    memset(buffer, 0, sizeof(buffer));
    if(recv(conn, buffer, sizeof(buffer), 0)==0)
        break;
}
close(conn);
close(server_sockfd);
```

ctrl+c 退出 server 后 client 可以检测到退出

```
[root@RHEL-zby 01]# ./tcp_server1 4000
连接成功 !
[ ]  
X root@RHEL-zby:/home/homework/1551265-000107/01 (ssh)
[root@RHEL-zby 01]# ./client 192.168.2.231 4000
连接成功 !
[ ]
```

ctrl+c 退出 client 后 server 可以监测到退出

```
[root@RHEL-zby 01]# ./tcp_server1 4000
连接成功 !
[root@RHEL-zby 01]# [ ]
× root@RHEL-zby:/home/homework/1551265-000107/01
[root@RHEL-zby 01]# ./client 192.168.2.231 4000
连接成功 !
^C
[root@RHEL-zby 01]# [ ]
```

kill-9 杀死进程的方法

server 端能检测到并退出

```
[root@RHEL-zby 02]# ./server 4000
连接成功 !
[root@RHEL-zby 02]# [ ]
× root@RHEL-zby:/home/homework/1551265-000107/02
[root@RHEL-zby 02]# ./client 192.168.2.231 4000
连接成功 !
已杀死
[root@RHEL-zby 02]# [ ]
```

client 端能检测到并退出

```
[root@RHEL-zby 01]# ./tcp_server1 4000
连接成功 !
已杀死
[root@RHEL-zby 01]# [ ]
× root@RHEL-zby:/home/homework/1551265-000107/01
[root@RHEL-zby 01]# ./client 192.168.2.231 4000
连接成功 !
[root@RHEL-zby 01]# [ ]
```

REUSEADDR 选项未设置时 会提示端口被占用 过大约半分钟正常

```
[root@RHEL-zby test]# ./server 4000
bind: Address already in use
```

加入这段代码后 设置 REUSEADDR

```
int main(int argc,char* argv[])
{
    //定义sockfd
    int server_sockfd = socket(AF_INET,SOCK_STREAM,0);
    int enable = 1;
    if (setsockopt(server_sockfd, SOL_SOCKET, SO_REUSEADDR, &enable, sizeof(int)) < 0)
        perror("setsockopt(SO_REUSEADDR) failed");
    //定义sockaddr_in
    int port=atoi(argv[1]);
    struct sockaddr_in server_sockaddr;
    server_sockaddr.sin_family = AF_INET;
```

之后退出再连接不会报错

```
[root@RHEL-zby test]# ./server 4000
^C
[root@RHEL-zby test]# ./server 4000
```

3.

打印如图所示

```
[root@RHEL-zby 02]# ./server 4000
连接成功 !
client IP:192.168.2.232  port:37490
^C
X  root@RHEL-zby:/home/homework/1551265-000107/01
[root@RHEL-zby 01]# ./client 192.168.2.231 4000
连接成功 !
^C
```

修改如图所示 传入的第一个参数为指定的 client 连接端口号

```
[root@RHEL-zby 02]# ./server 4000
连接成功 !
client IP:192.168.2.232 port:12345
[]

× root@RHEL-zby:/home/homework/1551265-000107/02 (ssh)
[root@RHEL-zby 02]# ./client 12345 192.168.2.231 4000
连接成功 !
```

4.

用 ifconfig 可以看到这里有三个 ip 地址

```
[root@RHEL-zby 03]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.2.231 netmask 255.255.255.0 broadcast 192.168.2.255
              inet6 fe80::5afa:2209:89b3:6073 prefixlen 64 scopeid 0x20<link>
              inet6 fe80::aa1e:1251:b612:af72 prefixlen 64 scopeid 0x20<link>
                ether 00:0c:29:fd:98:37 txqueuelen 1000 (Ethernet)
                  RX packets 25055 bytes 2361568 (2.2 MiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 12582 bytes 2514085 (2.3 MiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens33:0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.2.234 netmask 255.255.255.0 broadcast 192.168.2.255
              ether 00:0c:29:fd:98:37 txqueuelen 1000 (Ethernet)

ens38: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        ether 00:0c:29:fd:98:41 txqueuelen 1000 (Ethernet)
          RX packets 7561 bytes 660416 (644.9 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1 (Local Loopback)
          RX packets 1248 bytes 98384 (96.0 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 1248 bytes 98384 (96.0 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

运行得到预期结果

```
[root@RHEL-zby 03]# ./server 192.168.2.231 4000
IP:192.168.2.234
IP:192.168.2.231
IP:127.0.0.1
连接成功 !
client IP:192.168.2.232 port:37504
[]

× root@RHEL-zby:/home/homework/1551265-000107/03
[root@RHEL-zby 03]# ./client 192.168.2.231 4000
连接成功 !
[]
```

```
[root@RHEL-zby 03]# ./server 192.168.2.234 4000
IP:192.168.2.234
IP:192.168.2.231
IP:127.0.0.1
连接成功 !
client IP:192.168.2.232 port:35964
[]

× root@RHEL-zby:/home/homework/1551265-000107/03
[root@RHEL-zby 03]# ./client 192.168.2.234 4000
连接成功 !
[]
```

如果连接未绑定端口 则连接被拒绝

```
[root@RHEL-zby 03]# ./server 192.168.2.234 4000
IP:192.168.2.234
IP:192.168.2.231
IP:127.0.0.1
[]

× root@RHEL-zby:/home/homework/1551265-000107/03
[root@RHEL-zby 03]# ./client 192.168.2.231 4000
connect: Connection refused
[root@RHEL-zby 03]#
```

5.

read/write 函数

read 一次 20 字节 write 一次 30 字节

server 端分两次接收和输出 由于中间加了 1s 的延时 在 client 端接收到的信号
为 30 字节连续字符流

```
[root@RHEL-zby 04]# ./tcp_server4-1 4000
连接成功!
36753562912709360626
1879202375
92289736129319478450
3610632061
55476569374525474430
7868843149
20689266495048717272
2610615949
09177115977673656394
8129390885
09638561159848103044
4476317596
21785741859753883189
6433386048
88977643030925405946
9224775481

[root@RHEL-zby 04]# []
× root@RHEL-zby:/home/homework/1551265-000107/04
[root@RHEL-zby 04]# ./client1 192.168.2.231 4000
连接成功!
发送:367535629127093606261879202375
367535629127093606261879202375发送:922897361293194784503610632061
发送:554765693745254744307868843149
3610632061发送:206892664950487172722610615949
45254744307868843149发送:091771159776736563948129390885
206892664950487172722610615949发送:096385611598481030444476317596
发送:217857418597538831896433386048
8129390885发送:889776430309254059469224775481
^C
[root@RHEL-zby 04]# ]
```

client 端 write 每次 2 字节 server 端的 read 没有阻塞直接返回 read 函数没有读满输出的阻塞

```
[root@RHEL-zby 04]# ./tcp_server4-1 4000
连接成功 !
00
01
02
03
04
[]

× root@RHEL-zby:/home/homework/1551265-000107/04 (ssh)

[root@RHEL-zby 04]# ./client12 192.168.2.231 4000
连接成功 !
发送 :00
00发送 :01
01发送 :02
02发送 :03
03发送 :04
[]
```

recv/send

server 端主函数中的循环 recv 的参数设置为阻塞 可以等待接收字节数达到 maxbit

每次 client 发送 30 字节 server 端 read 一次读 20
server

```
while(1)
{
    memset(buffer,0,sizeof(buffer));
    recv(conn, buffer, maxbit,0);
    if(strcmp(buffer,"exit\n")==0)
        break;
    int i;
    for(i=0;i<sizeof(buffer);i++)
        cout<<buffer[i];
    cout<<endl;
    send(conn, buffer, sizeof(buffer), 0);
}
close(conn);
```

client

```
for(i=0;i<99;i++)
{
    for(j=0;j<sizeof(buf);j++)
        buf[j]=rand()%10+'0';
    send(sock_cli, buf, sizeof(buf),0); //发送
    cout<<"发送:";
    for(j=0;j<sizeof(buf);j++)
        cout<<buf[j];
    cout<<endl;
    sleep(1);
    recv(sock_cli, recvbuf, sizeof(recvbuf),0); //接收
    if(strcmp(recvbuf,"exit\n")==0)
        break;
    fputs(recvbuf, stdout);
    memset(recvbuf, 0, sizeof(recvbuf));
}
memset(buf, 0, sizeof(buf));
```

运行发现 server 还是分两次接收 但是返回 client 的数量有些异常
返回 10 字节 20 字节 30 字节循环

```
[root@RHEL-zby 04]# ./tcp_server4-2 4000
连接成功 !
36753562912709360626
1879202375
92289736129319478450
3610632061
55476569374525474430
7868843149
20689266495048717272
2610615949
09177115977673656394
8129390885
09638561159848103044
4476317596
21785741859753883189
6433386048
88977643030925405946
9224775481
28936802105110850646
2586284724
06299081311034039196
9338056640
[]

× root@RHEL-zby:/home/homework/1551265-000107/04

[root@RHEL-zby 04]# ./client2 192.168.2.231 4000
连接成功 !
发送 :367535629127093606261879202375
367535629127093606261879202375发送 :922897361293194784503610632061
发送 :554765693745254744307868843149
3610632061发送 :206892664950487172722610615949
45254744307868843149发送 :091771159776736563948129390885
206892664950487172722610615949发送 :096385611598481030444476317596
发送 :217857418597538831896433386048
8129390885发送 :889776430309254059469224775481
98481030444476317596发送 :289368021051108506462586284724
217857418597538831896433386048发送 :062990813110340391969338056640
```

client 一次 send 2 字节 server 端 recv20 字节

运行发现还是无阻塞 recv 到 2 字节即输出了

```
[root@RHEL-zby 04]# ./tcp_server4-2 4000
连接成功 !
00
01
02
03
04
05
06
07
```

```
[root@RHEL-zby 04]# ./client22 192.168.2.231 4000
连接成功 !
发送 :00
00发送 :01
01发送 :02
02发送 :03
03发送 :04
04发送 :05
05发送 :06
06发送 :07
^C
```

如果一定要让读入阻塞 不读满 20 不输出
recv 比较好办 直接把后面的参数改成 waitall 即可实现阻塞

```
while(1)
{
    memset(buffer,0,sizeof(buffer));
    recv(conn, buffer, maxbit,MSG_WAITALL);
    if(strcmp(buffer,"exit\n") == 0)
        break;
    cout<<buffer<<endl;
    send(conn, buffer, sizeof(buffer), 0);
}
```

```
[root@RHEL-zby 04]# ./server2-2 4000
连接成功!
00010203040506070809
[ ]  
X root@RHEL-zby:/home/homework/1551265-000107/04
[root@RHEL-zby 04]# ./client22 192.168.2.231 4000
连接成功!
发送:00
发送:01
发送:02
发送:03
发送:04
发送:05
发送:06
发送:07
发送:08
发送:09
00010203040506070809发送:10
[ ]
```

达到接收 20 字节输出一次的效果

如何修改 read 函数 在网上查到可以修改最低水位(缓冲区下限)

```
struct sockaddr_in client_addr;
socklen_t length = sizeof(client_addr);

int recv_min_size = 20;
setsockopt(server_sockfd, SOL_SOCKET, SO_RCVLOWAT, (void *)&recv_min_size, sizeof(int));
//成功返回非负描述字，出错返回-1
int conn = accept(server_sockfd, (struct sockaddr*)&client_addr, &length);
if(conn<0)
{}
```

在原代码中添加这样两行 即可达到效果 输出与上一种完全一致 不再截图

猜测这种方法也是用于 recv 函数 不需要设置最后一个参数为 waitall 也可以实现阻塞 验证后确实如此 不再截图

关于 read 和 recv 函数的区别 write 和 send 函数的区别

recv 和 send 有 4 个参数 最后一个参数设置为 0 时与 read 和 write 完全一样
但是 recv 后面的参数如果改为 MSG_WAITALL 即可一直阻塞直到指定的条件满

足，或是发生了错误。

- 1)当读到了指定的字节时，函数正常返回，返回值等于 len
 - 2)当读到了文档的结尾时,函数正常返回.返回值小于 len
 - 3)当操作发生错误时,返回 -1,且配置错误为相应的错误号(errno)
- 还有很多其他参数 需要时可以用 man 查看帮助获得
相对来说 recv 和 send 的使用要比 read 和 write 灵活很多

6.

```
[root@RHEL-zby 05]# ./server 4000
连接成功 !
[]

× root@RHEL-zby:/home/homework/1551265-000107/05 (ssh)

[root@RHEL-zby 05]# g++ -o client client01.cpp
[root@RHEL-zby 05]# ./client 192.168.2.231 4000
连接成功 !
发送字节数 :20000
发送字节数 :40000
发送字节数 :60000
发送字节数 :80000
发送字节数 :100000
发送字节数 :120000
发送字节数 :140000
发送字节数 :160000
发送字节数 :180000
发送字节数 :200000
发送字节数 :220000
发送字节数 :240000
发送字节数 :260000
发送字节数 :280000
发送字节数 :300000
发送字节数 :320000
发送字节数 :340000
发送字节数 :360000
发送字节数 :380000
```

server 端查看

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57187 ESTABLISHED
tcp     0 130320 RHEL-zby:37588           192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57139 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]#
```

client 端查看

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56586 ESTABLISHED
tcp 264800 0 RHEL-zby:terabase          192.168.2.232:37588 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57152 ESTABLISHED
[root@RHEL-zby ~]#
```

解释其中重要的参数

Recv-Q:

表示收到的数据已经在本地接收缓冲，但是还有多少没有被进程取走，recv()

即接收队列 Recv-Q 一直处于阻塞状态

Send-Q:

对方没有收到的数据或者说没有 Ack 的,还是本地缓冲区.

如果发送队列 Send-Q 不能很快的清零，可能是有应用向外发送数据包过快，

或者是对方接收数据包不够快

即正常状态 Recv-Q 和 Send-Q 都应该为 0

重新运行 在运行过程中进行了 4 次 netstat -t 操作 write 不再返回时再操作一次

server 端截图:

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp   60000      0 RHEL-zby:terabase    192.168.2.232:37590 ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:56586   ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:57152   ESTABLISHED

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp  160000      0 RHEL-zby:terabase    192.168.2.232:37590 ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:56586   ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:57152   ESTABLISHED

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp  220000      0 RHEL-zby:terabase    192.168.2.232:37590 ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:56586   ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:57152   ESTABLISHED

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp  264800      0 RHEL-zby:terabase    192.168.2.232:37590 ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:56586   ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:57152   ESTABLISHED

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp  264800      0 RHEL-zby:terabase    192.168.2.232:37590 ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:56586   ESTABLISHED
tcp      0      0 RHEL-zby:ssh        192.168.2.1:57152   ESTABLISHED
```

client 端截图:

```

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57187 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57139 ESTABLISHED
tcp     0      0 RHEL-zby:37590            192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57187 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57139 ESTABLISHED
tcp     0      0 RHEL-zby:37590            192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57187 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57139 ESTABLISHED
tcp     0      0 RHEL-zby:37590            192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57187 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57139 ESTABLISHED
tcp     0      55200 RHEL-zby:37590        192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57187 ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:57139 ESTABLISHED
tcp     0      130320 RHEL-zby:37590       192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh              192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]#

```

可以观察到开始时 client 端 Send-Q 一直为 0 即信息全部被 server 内核接收
 而 server 端的 Recv-Q 开始增加 之后保持最大值不变
 即开始时接收信息并阻塞在内核 没有被 read 接收 内核阻塞的 Recv-Q 达到其
 最大值(264800)时拒收 client 发来的信息
 client 端 Send-Q 开始增加直到增到其最大值(130320)停止发送
 write 发出的信息阻塞在内核 直到达最大值停止 write
 两个最大值之和为 395120 在 380000 到 400000 之间 符合预期
 即 264800 字节阻塞在 server 内核 剩下 130320 字节阻塞在 client 内核

任意给个 getchar 可以将阻塞打开 之后可以一直发下去

```
^C
[root@RHEL-zby 05]# vi server01.cpp
[root@RHEL-zby 05]# ./server 4000
连接成功！

root@RHEL-zby:/home/homework/1551265-000107/05

发送字节数 :196820000
发送字节数 :196840000
发送字节数 :196860000
发送字节数 :196880000
发送字节数 :196900000
发送字节数 :196920000
发送字节数 :196940000
发送字节数 :196960000
发送字节数 :196980000
发送字节数 :197000000
发送字节数 :197020000
发送字节数 :197040000
发送字节数 :197060000
发送字节数 :197080000
发送字节数 :197100000
发送字节数 :197120000
发送字节数 :197140000
发送字节数 :197160000
发送字节数 :197180000
发送字节数 :197200000
发送字节数 :197220000
发送字节数 :197240000
发送字节数 :197260000
```

多次运行 netstat -t 可以看出 server 端的 Recv-Q 一直为 0 而 client 端的 Send-Q 有小幅波动 但不会持续增加并阻塞 可能为硬件速度原因所致

```

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:terabase       192.168.2.232:37712 ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64549  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64155  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64553  ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:terabase       192.168.2.232:37712 ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64549  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64155  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64553  ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:terabase       192.168.2.232:37712 ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64549  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64155  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64553  ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:terabase       192.168.2.232:37712 ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64549  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64155  ESTABLISHED
tcp      0      0 RHEL-zby:ssh            192.168.2.1:64553  ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156  ESTABLISHED
tcp      0      36 RHEL-zby:ssh           192.168.2.1:64550  ESTABLISHED
tcp      0      0 RHEL-zby:37712         192.168.2.231:terabase ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156  ESTABLISHED
tcp      0      36 RHEL-zby:ssh           192.168.2.1:64550  ESTABLISHED
tcp      0      41176 RHEL-zby:37712       192.168.2.231:terabase ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156  ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64550  ESTABLISHED
tcp      0      904 RHEL-zby:37712         192.168.2.231:terabase ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156  ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64550  ESTABLISHED
tcp      0      0 RHEL-zby:37712         192.168.2.231:terabase ESTABLISHED

```

将代码修改成每次发送 20 个字节 重新进行试验

传输 14 多 w 字节 write 进入阻塞状态

```
发送字节数 :1467300  
发送字节数 :1467320  
发送字节数 :1467340  
发送字节数 :1467360  
发送字节数 :1467380  
发送字节数 :1467400  
发送字节数 :1467420  
发送字节数 :1467440  
发送字节数 :1467460
```

阻塞时用 netstat -t 查询可得其 Recv-Q 和刚才比基本不变 Send-Q 增大了不少

```
[root@RHEL-zby ~]# netstat -t      tcp      0 1147904 RHEL-zby:37612      192.168.2.  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Addr      Proto Recv-Q Send-Q Local Address  
tcp 0 0 RHEL-zby:s      tcp 0 0 RHEL-zby:ssh  
tcp 246184 0 RHEL-zby:t      192.168.2.  
tcp 0 0 RHEL-zby:s      tcp 0 0 RHEL-zby:ssh  
[root@RHEL-zby ~]# netstat -t      192.168.2.  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Addr      Proto Recv-Q Send-Q Local Address  
tcp 0 0 RHEL-zby:s      tcp 0 1130608 RHEL-zby:37614      192.168.2.  
tcp 0 0 RHEL-zby:ssh  
[root@RHEL-zby ~]# netstat -t      192.168.2.  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Addr      Proto Recv-Q Send-Q Local Address  
tcp 0 0 RHEL-zby:s      tcp 0 0 RHEL-zby:ssh  
tcp 249208 0 RHEL-zby:t      192.168.2.  
tcp 0 0 RHEL-zby:s      192.168.2.  
[root@RHEL-zby ~]# netstat -t      192.168.2.  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Addr      Proto Recv-Q Send-Q Local Address  
tcp 0 0 RHEL-zby:s      tcp 0 1213896 RHEL-zby:37618      192.168.2.  
tcp 0 0 RHEL-zby:ssh  
[root@RHEL-zby ~]# netstat -t      192.168.2.  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Addr      Proto Recv-Q Send-Q Local Address  
tcp 0 0 RHEL-zby:t      192.168.2.  
tcp 250172 0 RHEL-zby:s      192.168.2.  
tcp 0 0 RHEL-zby:s      192.168.2.  
tcp 0 0 RHEL-zby:s      [root@RHEL-zby ~]# netstat -t      192.168.2.  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Addr      Proto Recv-Q Send-Q Local Address  
tcp 0 0 RHEL-zby:s      192.168.2.  
tcp 247468 0 RHEL-zby:t      192.168.2.  
tcp 0 0 RHEL-zby:s      192.168.2.  
tcp 0 0 RHEL-zby:s      192.168.2.  
[root@RHEL-zby ~]# [root@RHEL-zby ~]#
```

打开阻塞后和之前情况一样 不再截图

调换 server 和 client 后

client 收信号 server 发信号 一次 50 字节无延时

```
发送字节数 :1403450
发送字节数 :1403500
发送字节数 :1403550
发送字节数 :1403600
发送字节数 :1403650
[×] root@RHEL-zby:/home/homework/1551265-000107/05
[root@RHEL-zby 05]# ./client 192.168.2.231 4000
连接成功 !
[×]
```

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0 1150840 RHEL-zby:terabase      192.168.2.232:37702 ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64549  ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64155  ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64553  ESTABLISHED
[root@RHEL-zby ~]# [×]
[×] root@RHEL-zby:~
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp    252850      0 RHEL-zby:37702      192.168.2.231:terabase ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64156  ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64550  ESTABLISHED
[root@RHEL-zby ~]# [×]
```

打开阻塞后 可以持续收发

```
发送字节数 :247330650  
发送字节数 :247330700  
发送字节数 :247330750  
发送字节数 :247330800  
发送字节数 :247330850  
发送字节数 :247330900  
发送字节数 :247330950  
发送字节数 :
```

```
× root@RHEL-zby:/home/homework/1551265-000107/05 (ssh)
```

```
[root@RHEL-zby 05]# ./client 192.168.2.231 4000  
连接成功！
```

```
a
```

```
□
```

且 Recv-Q 和 Send-Q 都基本为 0

231

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      2 RHEL-zby:terabase        192.168.2.232:37714 ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64549   ESTABLISHED
tcp      0  320344 RHEL-zby:ssh            192.168.2.1:64155   ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64553   ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:terabase        192.168.2.232:37714 ESTABLISHED
tcp      0      36 RHEL-zby:ssh            192.168.2.1:64549   ESTABLISHED
tcp      0  324348 RHEL-zby:ssh            192.168.2.1:64155   ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64553   ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:terabase        192.168.2.232:37714 ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64549   ESTABLISHED
tcp      0  117280 RHEL-zby:ssh            192.168.2.1:64155   ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64553   ESTABLISHED
[root@RHEL-zby ~]#
```

232

```
× root@RHEL-zby:~ (ssh)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156   ESTABLISHED
tcp      0      0 RHEL-zby:37714           192.168.2.231:terabase ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64550   ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156   ESTABLISHED
tcp      0      0 RHEL-zby:37714           192.168.2.231:terabase ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64550   ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156   ESTABLISHED
tcp      0      0 RHEL-zby:37714           192.168.2.231:terabase ESTABLISHED
tcp      0      36 RHEL-zby:ssh            192.168.2.1:64550   ESTABLISHED
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64156   ESTABLISHED
tcp      0      0 RHEL-zby:37714           192.168.2.231:terabase ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:64550   ESTABLISHED
[root@RHEL-zby ~]#
```

server 端加入设置接收缓冲区为 32k

```
int nRecvBuf = 32 * 1024; //设置为32K  
setsockopt(server_sockfd,SOL_SOCKET,S0_RCVBUF,(const char*)&nRecvBuf,sizeof(int));
```

client 端加入设置发送缓冲区为 32k

```
int nSendBuf = 32*1024; //设置为32K  
setsockopt(sock_cli,SOL_SOCKET,S0_SNDBUF,(const char*)&nSendBuf,sizeof(int));
```

重新运行

```
[root@RHEL-zby 05]# ./server 4000  
连接成功 !  
[]  
  
× root@RHEL-zby:/home/homework/1551265-000107/05 (ssh)  
连接成功 !  
发送字节数 :20000  
发送字节数 :40000  
发送字节数 :60000  
发送字节数 :80000  
发送字节数 :100000  
^C
```

手速不够快 只进行了两次查看操作

中间进行一次 netstat -t 操作 结束进行一次

server

```
[root@RHEL-zby ~]# netstat -t  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Address          Foreign Address      State  
tcp    40000      0 RHEL-zby:terabase    192.168.2.232:37600  ESTABLISHED  
tcp      0      0 RHEL-zby:ssh           192.168.2.1:56586   ESTABLISHED  
tcp      0      0 RHEL-zby:ssh           192.168.2.1:59587   ESTABLISHED  
tcp      0      0 RHEL-zby:ssh           192.168.2.1:57152   ESTABLISHED  
[root@RHEL-zby ~]# netstat -t  
Active Internet connections (w/o servers)  
Proto Recv-Q Send-Q Local Address          Foreign Address      State  
tcp    45792      0 RHEL-zby:terabase    192.168.2.232:37600  ESTABLISHED  
tcp      0      0 RHEL-zby:ssh           192.168.2.1:56586   ESTABLISHED  
tcp      0      0 RHEL-zby:ssh           192.168.2.1:59587   ESTABLISHED  
tcp      0      0 RHEL-zby:ssh           192.168.2.1:57152   ESTABLISHED  
[root@RHEL-zby ~]#
```

client

```

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address
tcp      0      0 RHEL-zby:ssh             192.168.2.1:57187
tcp      0      54208 RHEL-zby:37596      192.168.2.231:terabase
tcp      0      0 RHEL-zby:ssh             192.168.2.1:56589
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address
tcp      0      0 RHEL-zby:ssh             192.168.2.1:57187
tcp      0      70952 RHEL-zby:37596      192.168.2.231:terabase
tcp      0      0 RHEL-zby:ssh             192.168.2.1:56589
[root@RHEL-zby ~]# netstat -t

```

打开阻塞可以正常收发 和之前一样 不再截图

像上次一样把每次传输字节数设为 20 sleep(1)去掉

server

```

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:56586 ESTABLISHED
tcp      41124    0 RHEL-zby:terabase     192.168.2.232:37634 ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:57152 ESTABLISHED
[root@RHEL-zby ~]#

```

client

```

[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp      0      0 RHEL-zby:ssh             192.168.2.1:57187 ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:62054 ESTABLISHED
tcp      0      72400 RHEL-zby:37634      192.168.2.231:terabase ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:59825 ESTABLISHED
tcp      0      0 RHEL-zby:ssh             192.168.2.1:56589 ESTABLISHED
[root@RHEL-zby ~]#

```

```
发送字节数 :113360
发送字节数 :113380
发送字节数 :113400
发送字节数 :113420
发送字节数 :113440
发送字节数 :113460
发送字节数 :113480
发送字节数 :113500
发送字节数 :113520
```

和之前基本一样没有什么变化

7.

第一次:双收双发

连接成功后 server 发送 两边都阻塞在 read 函数

```
[root@RHEL-zby 06]# ./server6-1 4000 1000 1000
连接成功 !
[ ]  
× root@RHEL-zby:/home/homework/1551265-000107/06 (ssh)
[root@RHEL-zby 06]# ./client6-1 192.168.2.231 4000 1000 1000
连接成功 !
[ ]
```

查 netstat -t 当然也是没有数据堆积在内核

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64776  ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64549  ESTABLISHED
tcp     0      0 RHEL-zby:terabase      192.168.2.232:37724 ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64155  ESTABLISHED
[root@RHEL-zby ~]# []

× root@RHEL-zby:~
```

```
[root@RHEL-zby ~]# netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64156  ESTABLISHED
tcp     0      0 RHEL-zby:ssh            192.168.2.1:64550  ESTABLISHED
tcp     0      0 RHEL-zby:37724         192.168.2.231:terabase ESTABLISHED
[root@RHEL-zby ~]# ]
```

第二次:双发双收

双 1000

连接成功后 可以一直收发消息且收发消息字节数相近(硬件原因 理论相同)

```
反达子节数 :7705000
发送字节数 :7706000
发送字节数 :7707000
发送字节数 :7708000
发送字节数 :7709000
发送字节数 :7710000
发送字节数 :7711000
发送字节数 :7712000
[]

× root@RHEL-zby:/home/homework/1551
发送字节数 :7503000
发送字节数 :7504000
发送字节数 :7505000
发送字节数 :7506000
发送字节数 :7507000
发送字节数 :7508000
发送字节数 :7509000
发送字节数 :7510000
发送字节数 :7511000
发送字节数 :7512000
发送字节数 :7513000
发送字节数 :7514000
发送字节数 :7515000
发送字节数 :7516000
发送字节数 :7517000
发送字节数 :7518000
发送字节数 :7519000
发送字节数 :7520000
发送字节数 :7521000
发送字节数 :7522000
发送字节数 :7523000
发送字节数 :7524000
```

1000/500

连接成功后 可以一直收发 server 的发送是 client 的一半

```
发送字节数 :4954000
发送字节数 :4954500
发送字节数 :4955000
发送字节数 :4955500
发送字节数 :4956000
发送字节数 :4956500
[]

× root@RHEL-zby:/home/homewor
发送字节数 :9765000
发送字节数 :9766000
发送字节数 :9767000
发送字节数 :9768000
发送字节数 :9769000
发送字节数 :9770000
发送字节数 :9771000
发送字节数 :9772000
发送字节数 :9773000
发送字节数 :9774000
发送字节数 :9775000
发送字节数 :9776000
发送字节数 :9777000
发送字节数 :9778000
发送字节数 :9779000
发送字节数 :9780000
发送字节数 :9781000
发送字节数 :9782000
发送字节数 :9783000
发送字节数 :9784000
发送字节数 :9785000
发送字节数 :9786000
```

双 1000/双 700

两端正常收发

```
发送字节数 :56997000
发送字节数 :56998000
发送字节数 :56999000
发送字节数 :57000000
发送字节数 :57001000
发送字节数 :57002000
发送字节数 :57003000
发送字节数 :57004000
发送字节数 :56793100
发送字节数 :56793800
发送字节数 :56794500
发送字节数 :56795200
发送字节数 :56795900
发送字节数 :56796600
发送字节数 :56797300
发送字节数 :56798000
发送字节数 :56798700
发送字节数 :56799400
发送字节数 :56800100
发送字节数 :56800800
发送字节数 :56801500
发送字节数 :56802200
发送字节数 :56802900
发送字节数 :56803600
发送字节数 :56804300
发送字节数 :56805000
发送字节数 :56805700
发送字节数 :56806400
发送字节数 :56807100
发送字节数 :56807800
```

第三次实验:错开收发

双 1000

连接成功 可以持续收发 两端发送数相等

```
发送字节数 :7717000
发送字节数 :7718000
发送字节数 :7719000
发送字节数 :7720000
发送字节数 :7721000
发送字节数 :7722000
发送字节数 :7723000
□

× root@RHEL-zby:/home/home
发送字节数 :7640000
发送字节数 :7641000
发送字节数 :7642000
发送字节数 :7643000
发送字节数 :7644000
发送字节数 :7645000
发送字节数 :7646000
发送字节数 :7647000
发送字节数 :7648000
发送字节数 :7649000
发送字节数 :7650000
发送字节数 :7651000
发送字节数 :7652000
发送字节数 :7653000
发送字节数 :7654000
发送字节数 :7655000
发送字节数 :7656000
发送字节数 :7657000
发送字节数 :7658000
发送字节数 :7659000
发送字节数 :7660000
发送字节数 :7661000
```

1000/500

还是 client 是 server 发送量的两倍

```
发送字节数 :4251000
发送字节数 :4251500
发送字节数 :4252000
发送字节数 :4252500
发送字节数 :4253000
发送字节数 :4253500
发送字节数 :4254000
□
```

```
root@RHEL-zby:/home/home
发送字节数 :8394000
发送字节数 :8395000
发送字节数 :8396000
发送字节数 :8397000
发送字节数 :8398000
发送字节数 :8399000
发送字节数 :8400000
发送字节数 :8401000
发送字节数 :8402000
发送字节数 :8403000
发送字节数 :8404000
发送字节数 :8405000
发送字节数 :8406000
发送字节数 :8407000
发送字节数 :8408000
发送字节数 :8409000
发送字节数 :8410000
发送字节数 :8411000
发送字节数 :8412000
发送字节数 :8413000
发送字节数 :8414000
发送字节数 :8415000
```

双 1000/双 700

还是正常收发 两端相同

```
发送字节数 :31400000
发送字节数 :31407000
发送字节数 :31408000
发送字节数 :31409000
发送字节数 :31410000
发送字节数 :31411000
发送字节数 :31412000
发送字节数 :31413000
]
× root@RHEL-zby:/home/homework/
发送字节数 :31495100
发送字节数 :31495800
发送字节数 :31496500
发送字节数 :31497200
发送字节数 :31497900
发送字节数 :31498600
发送字节数 :31499300
发送字节数 :31500000
发送字节数 :31500700
发送字节数 :31501400
发送字节数 :31502100
发送字节数 :31502800
发送字节数 :31503500
发送字节数 :31504200
发送字节数 :31504900
发送字节数 :31505600
发送字节数 :31506300
发送字节数 :31507000
发送字节数 :31507700
发送字节数 :31508400
发送字节数 :31509100
发送字节数 :31509800
发送字节数 :31510500
```

第四次实验:错开收发(顺序颠倒)

双 1000

正常收发 两端相等

```
发送字节数:6735000
发送字节数:6736000
发送字节数:6737000
发送字节数:6738000
发送字节数:6739000
发送字节数:6740000
[]

× root@RHEL-zby:/home/1

发送字节数:6921000
发送字节数:6922000
发送字节数:6923000
发送字节数:6924000
发送字节数:6925000
发送字节数:6926000
发送字节数:6927000
发送字节数:6928000
发送字节数:6929000
发送字节数:6930000
发送字节数:6931000
发送字节数:6932000
发送字节数:6933000
发送字节数:6934000
发送字节数:6935000
发送字节数:6936000
发送字节数:6937000
发送字节数:6938000
发送字节数:6939000
发送字节数:6940000
发送字节数:6941000
发送字节数:6942000
[]
```

1000/500

正常收发 client 是 server 发送的两倍

```
发送字节数 :4877500
发送字节数 :4878000
发送字节数 :4878500
发送字节数 :4879000
发送字节数 :4879500
发送字节数 :4880000
[]

× root@RHEL-zby:/home/home1

发送字节数 :9738000
发送字节数 :9739000
发送字节数 :9740000
发送字节数 :9741000
发送字节数 :9742000
发送字节数 :9743000
发送字节数 :9744000
发送字节数 :9745000
发送字节数 :9746000
发送字节数 :9747000
发送字节数 :9748000
发送字节数 :9749000
发送字节数 :9750000
发送字节数 :9751000
发送字节数 :9752000
发送字节数 :9753000
发送字节数 :9754000
发送字节数 :9755000
发送字节数 :9756000
发送字节数 :9757000
发送字节数 :9758000
发送字节数 :9759000
[]
```

1000/700

正常收发 两端相等

```
发送字节数 :43738000
发送字节数 :43739000
发送字节数 :43740000
发送字节数 :43741000
发送字节数 :43742000
发送字节数 :43743000
发送字节数 :43744000
```

```
]
```

```
× root@RHEL-zby:/home/homewo
```

```
发送字节数 :42441700
发送字节数 :42442400
发送字节数 :42443100
发送字节数 :42443800
发送字节数 :42444500
发送字节数 :42445200
发送字节数 :42445900
发送字节数 :42446600
发送字节数 :42447300
发送字节数 :42448000
发送字节数 :42448700
发送字节数 :42449400
发送字节数 :42450100
发送字节数 :42450800
发送字节数 :42451500
发送字节数 :42452200
发送字节数 :42452900
发送字节数 :42453600
发送字节数 :42454300
发送字节数 :42455000
发送字节数 :42455700
发送字节数 :42456400
```

```
]
```