**计算机网络及安全期末作业报告**

**一、文件描述：**

pro.py：发送和接收函数的定义问件。

server.py：服务器main函数

client.py：客户端main函数

test：测试样本

**二、数据结构：**

class cube:  
 def \_\_init\_\_(self**,** s**,** n=**0**):  
 self.num = str(n) #seq  
 self.state = **0** #是否被发送self.isack = **0** #是否确认接收self.msg = s #数据内容  
  
 def \_\_str\_\_(self):  
 return self.num + ' ' + self.msg #发送报文格式：序列号 数据

**三、GBN部分：**

**1、服务器发送函数的思路：**

建立一个最大为10的窗口，每次循环用文件中的内容尽可能地将窗口装满。如果文件已读到末尾，且窗口为空，则说明所有数据均发送并接收成功了，结束测试。

如果检测到超时，就将窗口中的所有分组设为未发送状态，然后将窗口中的所有未发送状态的分组发送。

然后如果从客户端收到ack，计时器记为0，将序列号小于等于ack的分组从窗口中移除。

未收到分组，时间加一。

继续循环。

**2、客户端接收函数的思路：**

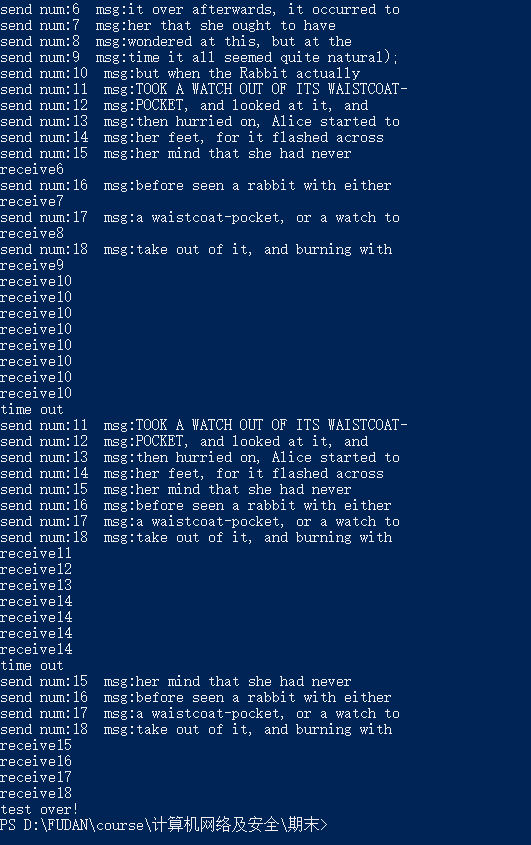
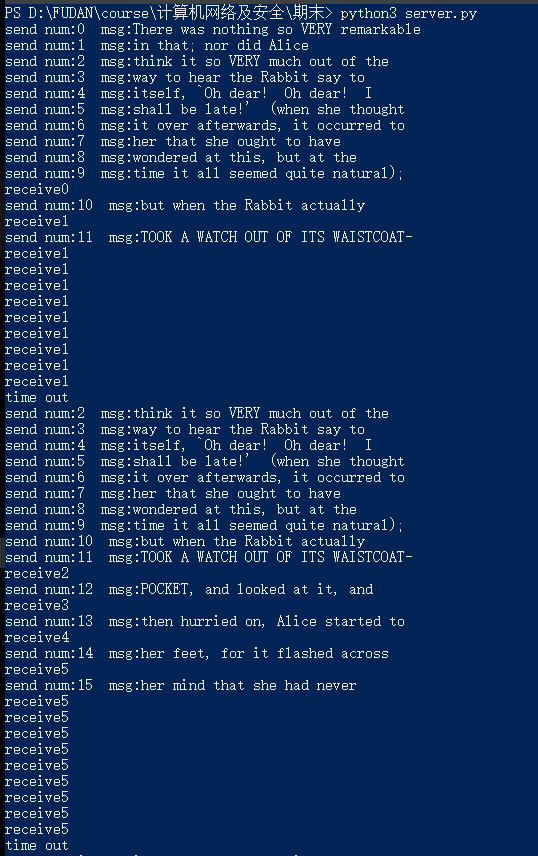
如果收到结束信号，结束测试。

收到了预期的分组，就向服务器发送确认信息，保存数据。

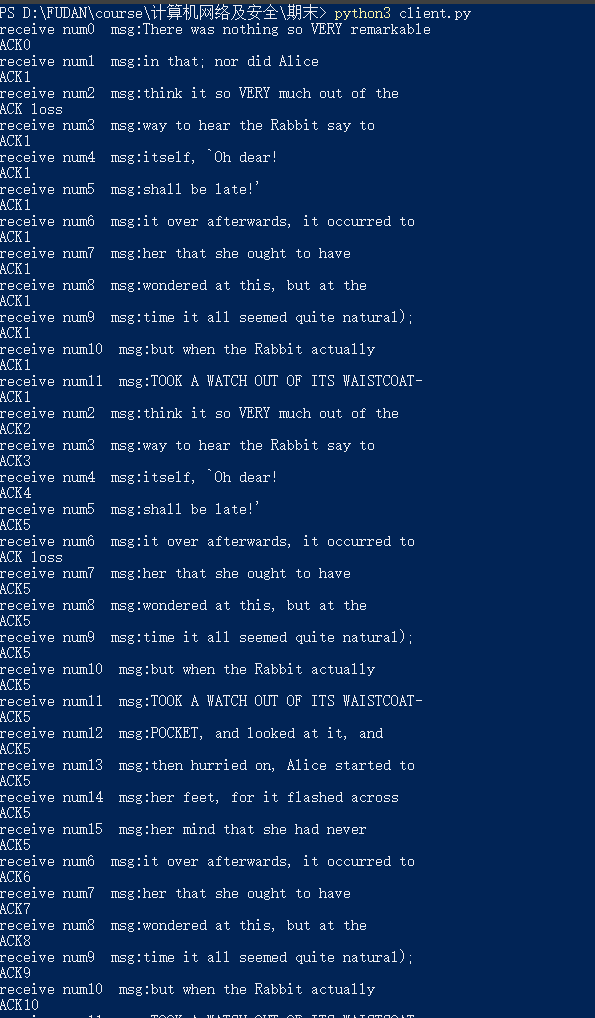
没有收到预期的分组，就向服务器发送上一个被确认的分组信息（如果没有上一组，就发送-1）。

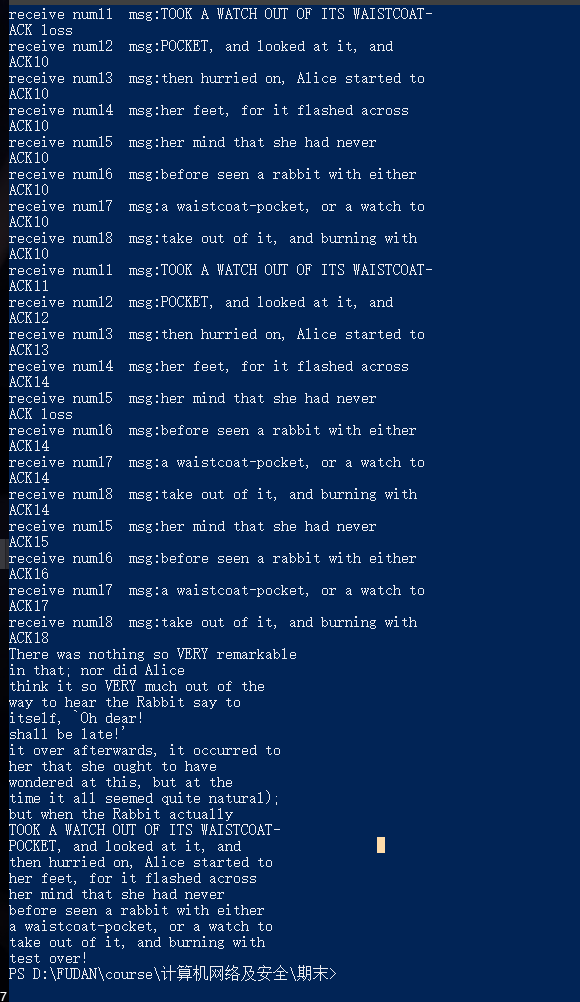
**3、实验结果：**

服务器：



客户端：





**四、SR部分：**

**1、服务器发送函数的思路：**

建立一个最大为10的窗口，每次循环用文件中的内容尽可能地将窗口装满。如果文件已读到末尾，且窗口为空，则说明所有数据均发送并接收成功了，结束测试。

如果检测到超时，就将窗口中的所有未被确认接收的分组设为未发送状态，然后将窗口中的所有未发送状态的分组发送。

然后如果从客户端收到ack，计时器重置为0，将第一个未被确认的分组之前的分组从窗口中移除。

未收到分组，时间加一。

继续循环。

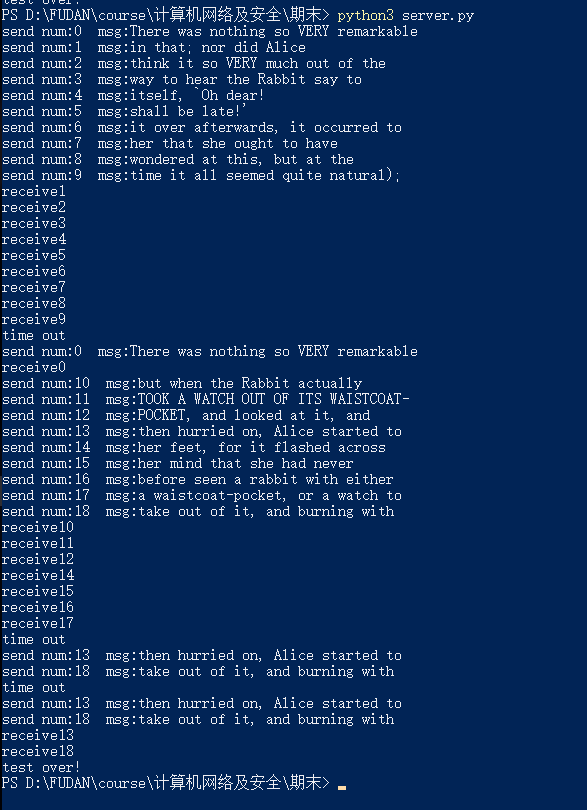
**2、客户端接收函数的思路：**

如果收到结束信号，结束测试。

只要收到了分组，就向服务器发送确认信息，保存分组，并按序列号将分组排序。

**3、实验结果：**

服务器：



客户端：

