

UDP文件传输程序

学号：19335018 姓名：陈俊熹

UDP文件传输程序

程序说明

使用说明

实现方法

程序测试

题目要求：写一个文件传输程序，通过 UDP 来传输一个文件及其管理元数据（文件名、大小和 日期等），在应用层处理报文出错、重组和乱序等问题，保证文件传输正确无误。

程序说明

使用说明

- 程序在 Linux(ubuntu)下运行。
- 提交的压缩包里有文件夹，一个是利用随机数模拟了糟糕的网络环境下经常丢包的 UDP 文件传输(udp_file)，可以在本地测试解决乱序和丢包的功能。另一个则是不带模拟，正常的 UDP 文件传输(udp_notest)，用于两个主机间的正常的传输测试。
- 打开其中一个文件夹后编译运行即可，有相应的输入提示。
- 接收端的初始化要输入两个端口，发送端的目标端口填入接收端初始化时的第一个端口（侦听端口）。

实现方法

- **接收端**有占用两个端口，一个用于监听传送请求，一个用于接收文件以及发送确认报文。
- **发送端**为单一端口，用于发送文件数据包和请求报文以及接收确认报文。
- 解决 UDP 的乱序以及丢包的方法为采用一问一答式的传输，即发送端每发送一个数据包后，只有接收到接收端的相应确认报文后才会开始发送下一个数据包。如果丢包太多，会终止传输。
- 数据包传送完成后，接收端会间隔发送冗余的确认语句，以对抗 UDP 可能的丢包；接收端收到确认语句后会退出，否则，会在多次发送无应答后，打印错误信息并退出。
- 数据的比特错误交由 UDP 协议自带的校验和解决，如有错误，则会转为解决丢包问题。
- 文件的属性(创建时间、修改时间、大小等)会在传输完成后打印。

程序测试

1. 无模拟丢包，直接传送（左边为发送程序，右侧为接收程序）

```
sanchen2@ubuntu:~/udp_notest$ ./send
Dest ip:192.168.152.128
Dest port:8080
filepath:/home/sanchen2/os_19335018_陈俊熹_labproject01.tar.xz
Filename:os_19335018_陈俊熹_labproject01.tar.xz
bind!
ip:192.168.152.128
port:8080
Dest ip:192.168.152.128
Dest port:8080
File:os_19335018_陈俊熹_labproject01.tar.xz
Get permission for sending
```

发送程序

接收程序

```
sanchen2@ubuntu:~/udp_notest$ ./recv
port for request:8080
port for recv:8081
request socket bind!
ip:192.168.152.128
port:8080
recv socket bind!
ip:192.168.152.128
port:8081
Waiting request
recvfrom(): Resource temporarily unavailable
Waiting request
```



```
Good sending:6
Good sending:7
Good sending:8
Good sending:9
Good sending:10
Good sending:11
Good sending:12
Good sending:13
Good sending:14
Good sending:15
Good sending:16
Good sending:17
Good sending:18
Good sending:19
Good sending:20
Good sending:21
Good sending:22
Good sending:23
Good sending:24
Good sending:25
Good sending:26
Good sending:27
Good sending:28
Good sending:29
Good sending:30
Good sending:31
Good sending:32
Good sending:33
Good sending:34
Good sending:35
Good sending:36
Good sending:37
Good sending:38
Good sending:39
Good sending:40
Sending ends successfully!
dest ip: ^C

recvfrom(): Resource temporarily unavailable
No ack from receiver for packet:21
Good sending:21
Good sending:22
recvfrom(): Resource temporarily unavailable
No ack from receiver for packet:24
Good sending:24
Good sending:25
Good sending:26
Good sending:27
Good sending:28
Good sending:29
Good sending:30
Good sending:31
recvfrom(): Resource temporarily unavailable
No ack from receiver for packet:32
Good sending:32
Good sending:33
Good sending:34
Good sending:35
Good sending:36
Good sending:37
Good sending:38
Good sending:39
Good sending:40
Sending ends successfully!
dest ip: ^C

good packet!:11
good packet!:12
good packet!:13
good packet!:14
good packet!:15
good packet!:16
good packet!:17
good packet!:18
good packet!:19
good packet!:20
good packet!:21
good packet!:22
good packet!:23
good packet!:24
half packet!:25
good packet!:25
good packet!:26
recvfrom(): Resource temporarily unavailable
good packet!:27
good packet!:28
good packet!:29
good packet!:30
good packet!:31
good packet!:32
good packet!:33
good packet!:34
good packet!:35
good packet!:36
good packet!:37
recvfrom(): Resource temporarily unavailable
half packet!:38
good packet!:38
good packet!:39
good packet!:40
Receive ALL!
filename:udp_file.zip
size:20607Bytes
Initial access time:2021-05-07 06:06:00
Initial modified time:2021-04-30 06:56:03
Initial change time:2021-04-30 06:56:03
Waiting request
recvfrom(): Resource temporarily unavailable
Waiting request
```

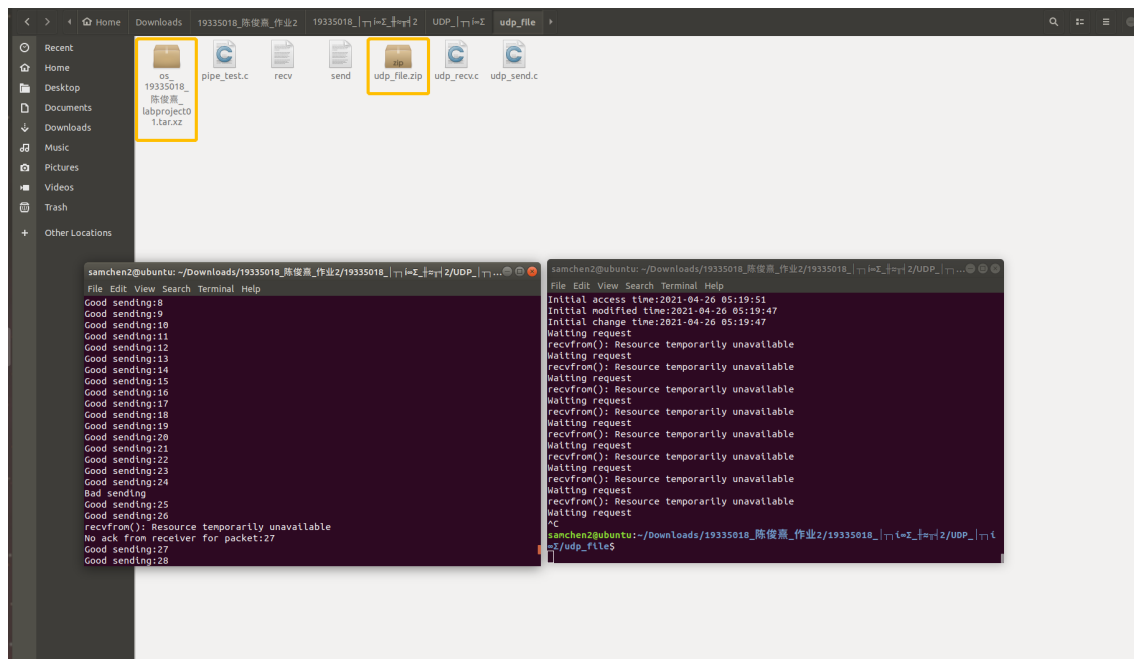
模拟网络险恶环境，丢包

模拟网络延时

重发

可以看到中间有一些断断续续，是模拟丢包和延迟起的作用，但最后还是完成了传输。

最终效果：



在本机上完成了从一个文件夹到另一个文件夹的传输（左右两个压缩包分别为无模拟传输和模拟丢包传输的结果）。