TCP文件传输程序

学号：19335286 姓名：郑有为

题目要求：写一个文件传输程序，通过一个TCP连接来传输一个文件及管理其元数据（包括文件名、文件格式、文件大小和日期等），保证文件传输正确无误。

程序说明

我们在Linux上用C语言编写了代码并跨虚拟机做了程序检验，整个TCP文件传输包括两个文件：`Client.c`和`Server.c`，前者是客户端代码，负责接收服务端上传的代码，后者是服务端代码，负责与客户端进行连接并发送文件数据。

发送的文件数据包括：文件的名称和格式、文件的大小、发送文件时的时间信息和文件本身的数据，文件格式不限。

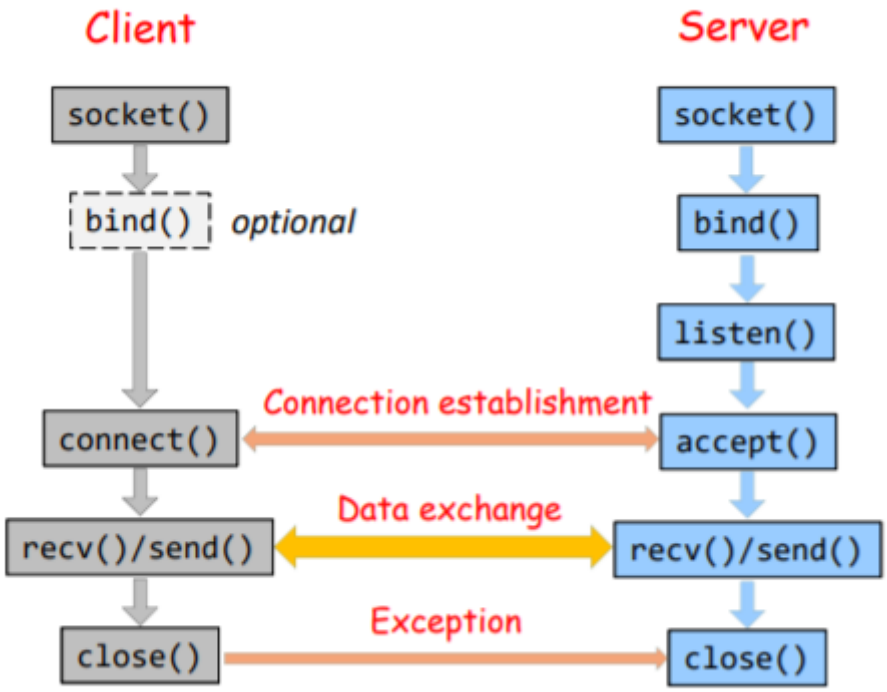
在我们的设计中，程序建立一个一对一的TCP连接，连接成功后传递一个文件，在传递结束后网络连接关闭并自动结束程序，在程序中，考虑TCP稳定的传输机制，未进行二次数据校验和丢包检测。

使用说明

- 建立连接：
  - 服务器：使用gcc编译并运行Server.c，此时该程序输出本地IP信息，堵塞等待客户端连接。
  - 客户机：使用gcc编译并运行Client.c，该程序会提示输入客户端IP地址，用户输入后等待连接。
- 传输文件：
  - 服务器：连接建立成功后，提示输入文件路径，若文件不存在或不可访问则返回错误，提示重新输入，若输入正确则开始向客户机发送数据包，发送结束后，返回发送完成并结束程序。
  - 客户机：堵塞等待服务机发送数据包，接收完毕后，打印文件信息并结束，用户可直接去程序所在的文件夹查看下载的文件。

程序原理

TCP网络编程建立连接的原理，参考下图：



实现细节

1. 报文的划分和打包
    - 服务端程序负责打包并发送文件，规定一个包的大小是1KB，对于比较大的文件，需要分多次传输。
    - 服务端负责获取文件信息并打包、拆分数据文件；客户端负责创建空文件，将收取数据包的信息写入文件。
    - 报文的格式如下：
        - 第一个报文包含文件信息和部分文件数据：
            - 第一行为文件名（包括文件类型），如 test.txt
            - 第二行为文件总大小，以字节为单位
            - 第三行为文件发送日期，其格式如：`Mon_Apr_19_17:22:34_2021`
            - 从第100个字节开始为文件数据
        - 其余的报文不包含文件信息，只包含文件数据。
2. 函数说明
    - 服务端程序函数：
        - `int get_time_string(char *t_str)`：

          负责获取当前的时间，并转化为字符串存，返回字符串地址到`t_str`中。
        - `int getipv4addr(char *ip_addr)`：

          自动从主机信息中搜索并输出网络IP信息，将对应的IP地址转化为字符串，返回字符串地址到`ip_addr`，如`192.168.73.129`。
        - `int file_choose(char *file_name)`：

          等待用户输入文件路径，然后对对应的文件进行检查（是否存在、是否可访问、文件大小），检查无误则从返回路径名供后续使用，返回值为文件大小。
        - 主函数：
            1. 负责创建并初始化`socket`，进行绑定，监听端口（以默认设定端口号），等待客户端发起连接请求。

2. 连接建立成功后，调用`file_choose`获取文件的各种信息，并创建文件句柄准备传输文件中的字符流。
3. 在While循环中，创建每一次传输的数据报文，往第一个报文加入文件的主要信息，文件名、大小等以便客户端创建文件副本，其余每次取文件里的1024字节，使用`send()`发送。
4. 发出空数据包以示发送结束，打印成功信息并断开连接、结束程序。

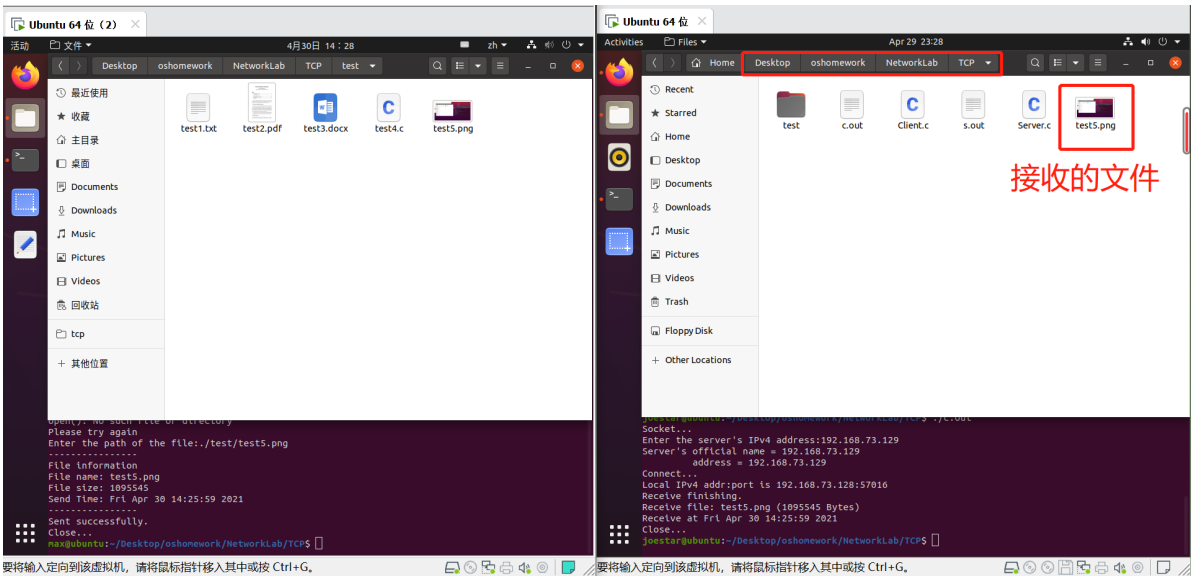
- 客户端服务程序：
  - 主函数：
    1. 自动检测本地IP，并等待用户请求服务器IP输入，然后创建并初始化`socket`，调用`connect`，尝试与服务器建立连接，连接建立后堵塞接收服务器数据包。
    2. 对接受的数据包进行计数，从第一个数据包中摘取文件名、文件大小、时间信息，在当前文件夹创建一个对应的同名文件和句柄，并写入数据包信息，不断接收数据包写入直至接收到空包（结束标识）。
    3. 最后输出接收的文件信息，断开连接，关闭文件句柄，结束程序。

程序测试

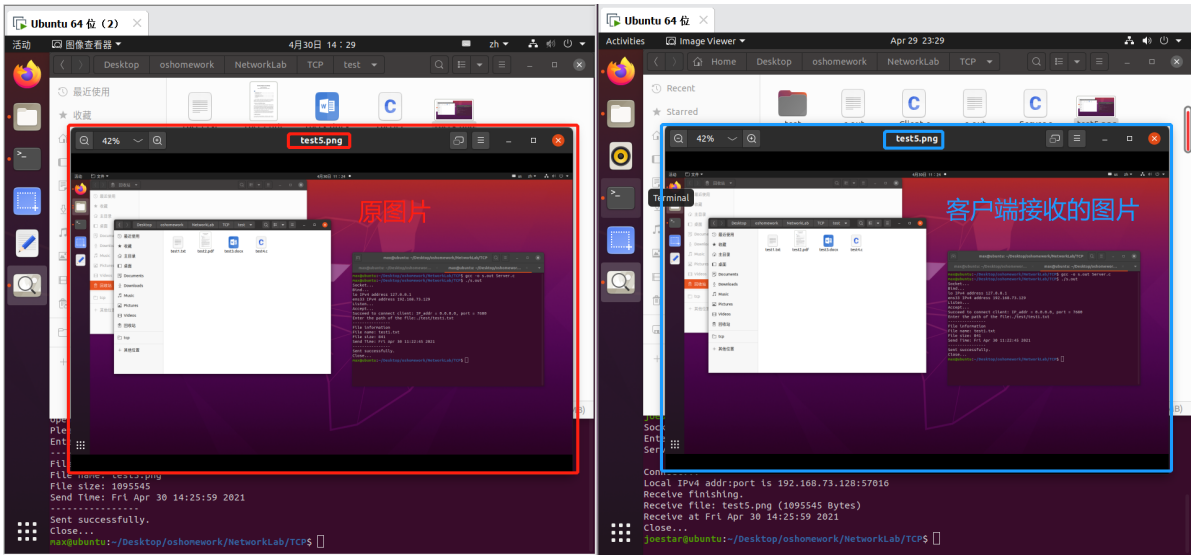
我们使用两台虚拟机进行测试，他们的IP地址分别是`192.168.73.129`和`192.168.73.128`，位于一个虚拟局域网中。

> 如图，启动程序，左侧是服务端（发文件），右侧是客户端（收文件），服务端向客户端发送了一张图片`test5.png`，客户端成功接收。

> 右侧图，可以看到下载的文件`test5.png`被保存在了`Client.c`所在的文件夹中。



> 下载的文件（右图）能正常打开并于原文件（左图）一致，无异常。



多次执行程序，下载不同类型的文件，可以看到文件都能正常传输。