

实例 2 TCP 文件下载

文件下载是指客户端将文件从服务器拷贝到本地。下载文件时，服务器将根据客户端输入的文件名到指定的目录中查找，若找到了相应的文件，服务器会读取文件，将读取的内容发送到客户端；客户端接收服务器发送的数据，提示用户选择下载的位置，并将接收到的数据写入到目标位置。

本实例要求编写程序，实现基于 TCP 的文件下载功能。

实例目标

- 掌握基于 TCP 的数据通信流程

实例描述

对实例进行分析，实例要求实现基于 TCP 的文件下载，其本质为实现基于 TCP 的数据通信与文件的读取和写入。TCP 文件下载分为 TCP 服务器端与 TCP 客户端两个程序，其中服务器端提供文件，客户端提供要下载的文件以及文件下载后存储的路径。

基于以上分析，这两个程序的功能分别为：

1. TCP 客户端：发起下载请求，将要下载的文件名及文件路径传送给 TCP 服务器，发起下载请求；接收服务器端返回的数据，在指定路径下创建文件，将服务器返回的文件内容写入文件并保存。
2. TCP 服务器：接收客户端请求，根据客户端发送来的文件名读取指定文件，将文件内容返回给 TCP 客户端。

代码实现

分别实现 TCP 服务器与 TCP 客户端，具体代码如下：

服务器.py

```
# 导入模块

import socket

import os, sys

# 创建 tcp 服务端 socket

tcp_server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 绑定端口

tcp_server_socket.bind(("", 8080))
```

```
# 设置监听，把服务端 socket 由主动套接字改成被动套接字，只能接收客户端的连接请求
tcp_server_socket.listen(128)

while True:

    # 接收客户端信息

    client_socket, client_ip = tcp_server_socket.accept()

    print("客户端: ", client_ip, "连接")

    # 接收下载信息

    file_name_data = client_socket.recv(1024)

    # 解码下载信息

    file_name = file_name_data.decode()

    # 判断文件是否存在

    try:

        with open(file_name, "rb") as file:

            while True:

                # 读取文件数据

                file_data = file.read(1024)

                # 数据长度不为 0 表示还有数据没有写入

                if file_data:

                    client_socket.send(file_data)

                # 数据为 0 表示传输完成

                else:

                    print(file_name, "传输成功")

                    break

    except Exception as e:

        print("传输异常: ", e)

    # 关闭客户端连接

    client_socket.close()
```

客户端.py

```
# 导入模块

import socket, os
```

```
# 创建套接字

tcp_client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 连接 IP 地址和端口

tcp_client_socket.connect(("127.0.0.1", 8080))

down_path = input("请输入文件路径: \n")

file_name = input("请输入文件名: \n")

all_path = os.path.join(down_path, file_name)

# 文件名编码

tcp_client_socket.send(all_path.encode())

# 判断文件是否存在

all_path = os.path.join(down_path, file_name)

try:

    all_files = os.listdir(down_path)

    if file_name in all_files:

        try:

            # 文件传输

            with open(os.getcwd() + '\\\\' + file_name, "wb") as file:

                while True:

                    # 接收数据

                    file_data = tcp_client_socket.recv(1024)

                    # 数据长度不为 0 写入文件

                    if file_data:

                        file.write(file_data)

                    # 数据长度为 0 表示下载完成

                    else:

                        break

            # 下载出现异常时捕获异常

        except Exception as e:

            print("下载异常", e)

    # 无异常则下载成功
```

```
else:

    print(file_name, "下载成功")

    # 关闭客户端

    tcp_client_socket.close()

else:

    print('指定文件不存在')

except FileNotFoundError as e:

    print('指定路径不存在')
```

以上实现的服务器和客户端符合 TCP 通信框架，其中服务器程序接收用户发送的请求信息（即要下载的文件），读取指定文件中的数据并将文件内容返回到客户端；客户端程序接收用户输入的文件名与文件存储路径，将文件名作为请求信息发送给服务器，根据存储路径和文件名在指定目录下创建或打开文件，并接收服务器返回的文件内容进行存储。

代码测试

先启动服务器端程序，再启动客户端程序。客户端启动后，连接建立成功，服务器端打印连接信息，客户端中打印提示信息，提示用户输入文件路径。文件路径输入完毕后，客户端再次打印提示信息，提示用户输入文件名。

1.若指定文件存在，文件将被下载到客户端程序所在目录中，此时服务器与客户端中打印的信息分别如下。

（1）服务器

```
客户端: ('127.0.0.1', 7581) 连接
E:\笔记\算法.txt 传输成功
```

（2）客户端

```
请输入文件路径:
E:\笔记
请输入文件名:
算法.txt
算法.txt 下载成功
```

2.若指定文件不存在，服务器端抛出异常，客户端提示“指定路径不存在”，具体如下所示。

（1）服务器

```
客户端: ('127.0.0.1', 7578) 连接
传输异常: [Errno 2] No such file or directory: '汉诺塔'
```

(2) 客户端

请输入文件路径:

请输入文件名:

汉诺塔

指定路径不存在