

# JEEK 信息社 2022 学年第二学期 期末任务

## 一、任务内容

实现一个文件传输客户端及服务端。

客户端需要进行如下两种请求：1) 上传文件 2) 下载文件

服务端需要对客户端的内容做出响应。

为了简化问题，我们假设只会有一个客户端与服务端连接。并且，我们不会对程序的鲁棒性做出特别的要求。你的程序应该运行在 11451 端口上。

## 二、协议内容

特别的，你的程序需要遵守 JFCP (Jianping File Communication Protocol) 协议（这样任何遵守协议的软件都可以交换使用）。JFCP 协议描述如下。

### 2.1 JFCP 请求（客户端）

一个请求包含三个部分。请求头、标头和消息主体（可选）。

#### 2.1.1 请求头

请求头由一行纯文本组成，以换行符 ('\n') 结尾。其中有两项，以空格分割：

- 请求的方法（GET 和 PUT，分别对应下载和上传文件）
- 要请求的文件（或要上传的文件）的路径（只需要考虑绝对路径）

#### 2.1.2 标头

标头用于传递额外信息。一个 JFCP 标头由它的名称（不用区分大小写）后跟随一个冒号 (:), 冒号后跟随它具体的值。

在这里，你只需要考虑一个标头：(Content-Length)，用于提示消息主体的长度，且只需在 PUT 请求时提供，语法如下

```
Content-Length: <length>
```

<length>为消息主体的长度，用十进制数字表示的字节的数目。

特别的，标头以一行空行结束 ('\n')

#### 2.1.3 消息主体

消息主体只在请求方法为 PUT 时提供。为文件的内容。

#### 2.1.4 示例

以下是一些合法的请求示例：

```
GET C:\path\foo.txt
```

下载 C:\path\foo.txt

```
PUT C:\upload\hello.txt
Content-Length: 381
```

```
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Cupiditate,
exercitationem totam quam ut velit delectus ea? Pariatur nobis aut
perspiciatis eligendi, voluptatum voluptatem quibusdam modi error
doloremque eaque commodi veritatis, in incidunt natus fugiat!
Accusantium neque eveniet quasi atque. Veniam soluta animi sed eius
enim, cumque ea? Quaerat, tenetur! Consequuntur!
```

上传到 C:\upload\hello.txt，文件大小为 381 字节，文件内容如消息主体所示。

## 2.2 JFSP 响应（服务端）

一个响应包含三个部分。响应头、标头和消息主体（可选）。

### 2.2.1 响应头

响应头由状态码组成，以换行符（'\n'）结尾。由于这是一个简化的协议，你只需要考虑两种状态码：

- 200 代表成功（对于 GET 代表返回了信息，对于 PUT 代表成功创建了文件）
- 300 代表失败（比如：GET 的文件不存在，PUT 未成功创建文件）

### 2.2.2 标头

响应也只有一个标头：（Content-Length），用于提示消息主体的长度，且只需在回应 GET 请求时提供，语法同上

```
Content-Length: <length>
```

<length>为消息主体的长度，用十进制数字表示的字节的数目。

### 2.2.3 消息主体

消息主体只在回应 GET 请求时提供。为文件的内容。

### 2.2.4 示例

以下是一些合法的响应示例：

```
200
Content-Length: 381
```

```
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Cupiditate,
exercitationem totam quam ut velit delectus ea? Pariatur nobis aut
perspiciatis eligendi, voluptatum voluptatem quibusdam modi error
doloremque eaque commodi veritatis, in incidunt natus fugiat!
Accusantium neque eveniet quasi atque. Veniam soluta animi sed eius
enim, cumque ea? Quaerat, tenetur! Consequuntur!
```

（GET 请求）成功找到要下载的文件，长 381 字节，内容如上。

200

(PUT 请求) 成功上传文件。

300

请求失败。

### 三、任务细节

#### 3.1 客户端代码

客户端代码执行逻辑如下：

- 输入 ip 地址和端口，连接到服务器
- 进入命令输入循环，此时应输出三个大于号 (>>>) 提示用户输入命令，命令有如下几种
  - GET <PATH\_LOCAL> <PATH\_REMOTE>:  
获取远程 PATH\_REMOTE 的文件，保存在 PATH\_LOCAL 的位置。
  - PUT <PATH\_LOCAL> <PATH\_REMOTE>:  
将本地 PATH\_LOCAL 的文件上传到远程 PATH\_REMOTE 的位置。
- 对每个命令，你都应该输出一行进行回复，若命令执行成功，输出 **Success!**，否则输出 **Failed!** 此外，对于 PUT 命令，如果本地的文件不存在，应该输出 **File not found!**，并且不会向服务端发送请求。

以下是一个示例的客户端代码执行结果（绿色斜体为用户输入）

```
Please input server IP: 127.0.0.1
Please input server port: 4433
>>> GET D:\result.png C:\apple.png
Success!
>>> GET D:\result.png C:\apple.pnf
Failed!
>>> PUT D:\result.png C:\apple2.png
Success!
>>> PUT D:\resultff.png C:\apple2.png
File not found!
```

#### 3.2 服务端代码

服务端代码不需要有输出，只需按照协议对每个请求进行回应即可（你只需要考虑一个客户端进行连接）。