

# 计算机网络 Lab3

---

## 一、实验目的

---

1. 通过 Python socket 编程建立 TCP 连接，掌握使用 Socket 创建网络连接的基本方法；
2. 学习如何发送自定义的应用层协议报文，了解应用层协议的构建和传输过程；
3. 使用 Wireshark 插件识别和观察自定义的应用层协议报文，探索网络工具的更多用法；
4. 深化对TCP连接、应用层协议和网络通信过程的理解，提高网络编程和调试技能。

## 二、实验任务

---

### 1.代码解释

`client.py`：首先设置客户端 `addr = ('localhost', 12000)` 定义服务器的地址和端口，`buffer_size = 1024` 设置数据缓冲区的大小为 1024 字节，用于接收和发送数据。然后创建 TCP 套接字，使用 `socket(AF_INET, SOCK_STREAM)` 创建 TCP 连接，并通过 `c.connect(addr)` 连接到服务器。然后进入发送数据的主循环：

- 客户端等待用户输入。如果输入 `#quit`，则跳出循环并关闭连接。
- 使用 `datetime.now(timezone.utc)` 获取当前的 GMT 时间并格式化。
- 构建自定义的应用层协议 FDUnet 报文，并将其通过 `c.sendall(data.encode())` 发送到服务器。
- 接收服务器的响应并打印出来

最后当用户输入 `#quit` 时，客户端关闭与服务器的连接，结束程序。

`server.py` : `servergo()` 函数用于处理每个客户端的连接，负责接收、处理和响应客户端的消息。在接收消息后，将其拆分为不同的部分（报文头、时间和实体内容）。然后检查消息实体内容是否为空，如果是空，则返回 `501 Not Implemented`。如果内容不为空，服务器将实体中的大写字母转换为小写，小写字母转换为大写，然后返回 `200 OK`，并附带转换后的字符串。当客户端发送 `#quit` 时，服务器将断开该客户端的连接，但服务器本身继续运行并服务其他客户端。

主循环：

- 服务器通过 `s.bind(addr)` 将套接字绑定到指定的地址和端口，并通过 `s.listen(5)` 开始监听连接请求。
- 当一个客户端连接时，服务器通过 `Thread` 创建一个新线程来处理该客户端的请求，确保多个客户端可以同时连接并通信。
- 在主线程中，服务器会继续等待新的客户端连接，直到终止服务器。

## 2.运行效果截图

1. 服务器分别对不同输入返回 `200 OK` 和 `501 Not Implemented`;

```
3> python .\client.py
>>input:AaBbCc
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 05:53:04 GMT
```

```
aAbBcC
>>input:
>> 1.0 501 Not Implemented
Date: Fri, 11 Oct 2024 05:53:08 GMT
```

```
>>input:[]
```

2. 一个服务器与一个客户端建立连接，并连续发送多条消息；

```
>>input:Hello,world!
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 05:55:06 GMT
```

```
hELLO,WORLD!
>>input:Oops~!
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 05:55:30 GMT
```

```
oOPS~!
>>input:GooD-Job1
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 05:55:56 GMT
```

```
gOOd-jOB1
>>input:█
```

### 3. 一个服务器同时与两个客户端建立连接，并发送消息；

```
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> python -u "c:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3\server.py"
listening on port 12000
connecting to ('127.0.0.1', 65527)
listening on port 12000
connecting to ('127.0.0.1', 65163)
listening on port 12000
[]
```

```
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> python .\client.py

>>input:Here is Client1!
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:01:14 GMT

HERE IS cCLIENT1!
>>input:Hello,Client1
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:01:38 GMT

HELLO,cCLIENT1
>>input:[]
```

```
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> python .\client.py
>>input:Here is Client2!
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:01:23 GMT

HERE IS cCLIENT2!
>>input:Hello,Client2
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:01:53 GMT

HELLO,cCLIENT2
>>input:[]
```

### 4. 任一客户端输入 #quit 后，双方关闭连接，但服务器继续服务其他客户端；

```
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> python -u "c:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3\server.py"
waiting for connection
connecting to addr ('127.0.0.1', 65257)
waiting for connection
connecting to addr ('127.0.0.1', 65259)
waiting for connection
Client disconnected.
```

```
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> python .\client.py

>>input:hello1
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:05:58 GMT

HELLO1
>>input:I Can Still SPEAK!
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:06:27 GMT

i cAN sTILL speak!
>>input:[]
```

```
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> python .\client.py
>>input:HELLO2
>> 1.0 200 OK
Date: Fri, 11 Oct 2024 06:06:03 GMT

hello2
>>input:#quit
PS C:\Users\IScream\Desktop\study\计算机网络\2024实验\hyf-22300240016-lab3> []
```

### 5. 选择 Wireshark 抓取到的任意一对 FDUnet 协议请求和应答报文，展示解析出的报文内容。

请求报文：

No.	Time	Source	Destination	Protocol	Length	Info
2588	54.466721	127.0.0.1	127.0.0.1	FDUnet	101	57060 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=57
2590	54.467017	127.0.0.1	127.0.0.1	FDUnet	101	12000 → 57060 [PSH, ACK] Seq=1 Ack=58 Win=2161152 Len=57
2609	58.507370	127.0.0.1	127.0.0.1	FDUnet	95	57060 → 12000 [PSH, ACK] Seq=58 Ack=58 Win=2161152 Len=51
2611	58.507614	127.0.0.1	127.0.0.1	FDUnet	108	12000 → 57060 [PSH, ACK] Seq=58 Ack=109 Win=2161152 Len=64
6413	176.589153	127.0.0.1	127.0.0.1	FDUnet	107	57060 → 12000 [PSH, ACK] Seq=109 Ack=122 Win=2161152 Len=63
6415	176.589488	127.0.0.1	127.0.0.1	FDUnet	107	12000 → 57060 [PSH, ACK] Seq=122 Ack=172 Win=2161152 Len=63
6752	200.135673	127.0.0.1	127.0.0.1	FDUnet	101	57060 → 12000 [PSH, ACK] Seq=172 Ack=185 Win=2160896 Len=57
6754	200.135844	127.0.0.1	127.0.0.1	FDUnet	101	12000 → 57060 [PSH, ACK] Seq=185 Ack=229 Win=2160896 Len=57
7485	226.578633	127.0.0.1	127.0.0.1	FDUnet	104	57060 → 12000 [PSH, ACK] Seq=229 Ack=242 Win=2160896 Len=60
7487	226.578865	127.0.0.1	127.0.0.1	FDUnet	104	12000 → 57060 [PSH, ACK] Seq=242 Ack=289 Win=2160896 Len=60
12732	380.219697	127.0.0.1	127.0.0.1	FDUnet	111	57060 → 12000 [PSH, ACK] Seq=289 Ack=302 Win=2160896 Len=67
12734	380.219922	127.0.0.1	127.0.0.1	FDUnet	111	12000 → 57060 [PSH, ACK] Seq=302 Ack=356 Win=2160896 Len=67
13286	395.718524	127.0.0.1	127.0.0.1	FDUnet	95	57175 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=51
13288	395.718743	127.0.0.1	127.0.0.1	FDUnet	108	12000 → 57175 [PSH, ACK] Seq=1 Ack=52 Win=2161152 Len=64
13360	407.412546	127.0.0.1	127.0.0.1	FDUnet	111	57175 → 12000 [PSH, ACK] Seq=52 Ack=65 Win=327168 Len=67
13362	407.412758	127.0.0.1	127.0.0.1	FDUnet	111	12000 → 57175 [PSH, ACK] Seq=65 Ack=119 Win=2161152 Len=67
13560	415.677422	127.0.0.1	127.0.0.1	FDUnet	111	57175 → 12000 [PSH, ACK] Seq=119 Ack=132 Win=327168 Len=67
13562	415.677608	127.0.0.1	127.0.0.1	FDUnet	111	12000 → 57175 [PSH, ACK] Seq=132 Ack=186 Win=2160896 Len=67
14656	463.802405	127.0.0.1	127.0.0.1	FDUnet	112	57213 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=68
14658	463.802637	127.0.0.1	127.0.0.1	FDUnet	112	12000 → 57213 [PSH, ACK] Seq=1 Ack=69 Win=2161152 Len=68
15426	505.957212	127.0.0.1	127.0.0.1	FDUnet	112	58996 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=68
15428	505.957472	127.0.0.1	127.0.0.1	FDUnet	112	12000 → 58996 [PSH, ACK] Seq=1 Ack=69 Win=2161152 Len=68
16126	544.652491	127.0.0.1	127.0.0.1	FDUnet	112	65527 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=68

Frame 6752: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface \Device\NPF\_{...} Loopback, id 0

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 57060, Dst Port: 12000, Seq: 172, Ack: 185, Len: 57

FDUnet protocol, used in computer net course

Firstline: Post / 1.0\n\n

[header]

Date: Fri, 11 Oct 2024 05:55:30 GMT

Data: Oops~!

应答报文：

No.	Time	Source	Destination	Protocol	Length	Info
2588	54.466721	127.0.0.1	127.0.0.1	FDUnet	101	57060 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=57
2590	54.467017	127.0.0.1	127.0.0.1	FDUnet	101	12000 → 57060 [PSH, ACK] Seq=1 Ack=58 Win=2161152 Len=57
2609	58.507370	127.0.0.1	127.0.0.1	FDUnet	95	57060 → 12000 [PSH, ACK] Seq=58 Ack=58 Win=2161152 Len=51
2611	58.507614	127.0.0.1	127.0.0.1	FDUnet	108	12000 → 57060 [PSH, ACK] Seq=58 Ack=109 Win=2161152 Len=64
6413	176.589153	127.0.0.1	127.0.0.1	FDUnet	107	57060 → 12000 [PSH, ACK] Seq=109 Ack=122 Win=2161152 Len=63
6415	176.589488	127.0.0.1	127.0.0.1	FDUnet	107	12000 → 57060 [PSH, ACK] Seq=122 Ack=172 Win=2161152 Len=63
6752	200.135673	127.0.0.1	127.0.0.1	FDUnet	101	57060 → 12000 [PSH, ACK] Seq=172 Ack=185 Win=2160896 Len=57
6754	200.135844	127.0.0.1	127.0.0.1	FDUnet	101	12000 → 57060 [PSH, ACK] Seq=185 Ack=229 Win=2160896 Len=57
7485	226.578633	127.0.0.1	127.0.0.1	FDUnet	104	57060 → 12000 [PSH, ACK] Seq=229 Ack=242 Win=2160896 Len=60
7487	226.578865	127.0.0.1	127.0.0.1	FDUnet	104	12000 → 57060 [PSH, ACK] Seq=242 Ack=289 Win=2160896 Len=60
12732	380.219697	127.0.0.1	127.0.0.1	FDUnet	111	57060 → 12000 [PSH, ACK] Seq=289 Ack=302 Win=2160896 Len=67
12734	380.219922	127.0.0.1	127.0.0.1	FDUnet	111	12000 → 57060 [PSH, ACK] Seq=302 Ack=356 Win=2160896 Len=67
13286	395.718524	127.0.0.1	127.0.0.1	FDUnet	95	57175 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=327424 Len=51
13288	395.718743	127.0.0.1	127.0.0.1	FDUnet	108	12000 → 57175 [PSH, ACK] Seq=1 Ack=52 Win=2161152 Len=64
13360	407.412546	127.0.0.1	127.0.0.1	FDUnet	111	57175 → 12000 [PSH, ACK] Seq=52 Ack=65 Win=327168 Len=67
13362	407.412758	127.0.0.1	127.0.0.1	FDUnet	111	12000 → 57175 [PSH, ACK] Seq=65 Ack=119 Win=2161152 Len=67
13560	415.677422	127.0.0.1	127.0.0.1	FDUnet	111	57175 → 12000 [PSH, ACK] Seq=119 Ack=132 Win=327168 Len=67
13562	415.677608	127.0.0.1	127.0.0.1	FDUnet	111	12000 → 57175 [PSH, ACK] Seq=132 Ack=186 Win=2160896 Len=67
14656	463.802405	127.0.0.1	127.0.0.1	FDUnet	112	57213 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=68
14658	463.802637	127.0.0.1	127.0.0.1	FDUnet	112	12000 → 57213 [PSH, ACK] Seq=1 Ack=69 Win=2161152 Len=68
15426	505.957212	127.0.0.1	127.0.0.1	FDUnet	112	58996 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=68
15428	505.957472	127.0.0.1	127.0.0.1	FDUnet	112	12000 → 58996 [PSH, ACK] Seq=1 Ack=69 Win=2161152 Len=68
16126	544.652491	127.0.0.1	127.0.0.1	FDUnet	112	65527 → 12000 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=68

Frame 6754: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface \Device\NPF\_{...} Loopback, id 0

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 12000, Dst Port: 57060, Seq: 185, Ack: 229, Len: 57

FDUnet protocol, used in computer net course

Firstline: 1.0 200 OK\n\n

[header]

Date: Fri, 11 Oct 2024 05:55:30 GMT

Data: oOPS~!

