## 作业2报告

1. 题目内容

使用UDP协议设计一个字典查询服务协议，客户端发送要查询英文单词给服务器，服务器将对应的中文含义送给客户端，客户端显示出中文含义。  
 字典库参见dict.zip

例如：  
 $ python Hw2.py server  
 $ python Hw2.py client accord  
 accord: v．同意，给予

$ python Hw2.py client noword  
 noword: 查询不到

1. 开发过程

包括下面几部分：

* + - 1. 系统结构，C/S，通信过程（请求，应答）

系统结构大概就是传统的C/S架构，client来发送请求，server接受并处理信息，在整个通信过程中，使用UDP协议进行通信，测试时用本机进行测试，整个系统由python开发，基于socket类库进行服务端以及客户端通信，并基于argparse来进行参数解析。

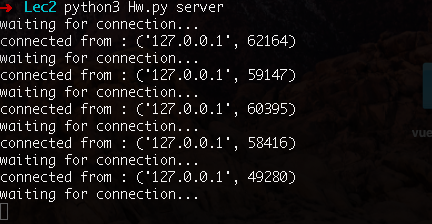
* + - 1. 通信协议

基于UDP协议进行开发，服务端接受编码方式为utf-8的英文单词字节串并进行解析，返回GBK编码方式的中文译文字节串，由client进行解析以及最终的解释。

1. 系统测试

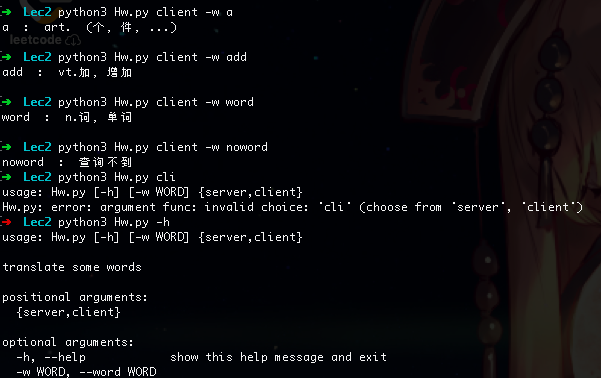
客户服务器测试的测试用例、流程及截图。(要求多一些测试用例)

server端测试



client端测试

(ps:测试包括正确单词，查询不到的单词，-h，参数错误等)



1. 心得体会

学习了通过python socket在server于client间进行通信，并学习了通过argparse来对终端参数进行解析以及错误处理，udp作为一个非面向对象连接的一个协议，与TCP相比操作简单了很多，在很多方面有着不一样的作用。

1. 源码

**项目结构**

.

├── Hw.py

└── dict

└── dict3.txt

1 directory, 2 files

Hw.py

1. #!/usr/bin/env python
2. # -\*- coding: utf-8 -\*-
3. """
4. \_\_title\_\_ = ''
5. \_\_author\_\_ = 'ngc7293'
6. \_\_mtime\_\_ = '2018/10/9'
7. # code is far away from bugs with the god animal protecting
8. I love animals. They taste delicious.
9. """
10. **from** socket **import** \*
11. **from** time **import** ctime
12. **from** argparse **import** ArgumentParser
14. **def** build\_parser():
15. parser = ArgumentParser(description="translate some words")
16. parser.add\_argument("func",choices = ['server', 'client'])
17. parser.add\_argument("-w","--word",dest="word",required=False)
18. **return** parser
20. **def** get\_dict():
21. dict\_file = open('dict/dict3.txt','rb')
22. dict\_word = {}
24. word\_lines = dict\_file.readlines()
26. **for** i **in** word\_lines:
27. str = i.decode('GBK')
28. word = str.split('   ')[0]
29. tran = str.split('   ')[1]
30. dict\_word[word] = tran
31. **return** dict\_word

34. **def** server():
35. dict\_word = get\_dict()
37. HOST = ""
38. PORT = 21567
39. BUFSIZ = 1024
40. ADDR = (HOST, PORT)
42. udpSerSock = socket(AF\_INET, SOCK\_DGRAM)
43. udpSerSock.bind(ADDR)
45. **while** True:
46. **print**("waiting for connection...")
47. data,addr = udpSerSock.recvfrom(BUFSIZ)
48. **print**("connected from :", addr)
50. **if** **not** data:
51. **break**
52. data = data.decode('utf-8')
54. **if** data **in** dict\_word.keys():
55. content = dict\_word[data]
56. **else**:
57. content = "error"
59. udpSerSock.sendto(content.encode("utf-8"),addr)
61. **def** client(word):
62. HOST = "127.0.0.1"
63. PORT = 21567
64. BUFSIZ = 1024
65. ADDR = (HOST, PORT)
67. udpCliSock = socket(AF\_INET, SOCK\_DGRAM)
68. udpCliSock.connect(ADDR)

71. data = word
73. udpCliSock.send(data.encode("utf8"))
74. data = udpCliSock.recv(BUFSIZ).decode('utf-8')
76. **if** data == "error":
77. data = "查询不到"
79. **print**(word," : ",data)
81. udpCliSock.close()
83. **def** main():
84. parser = build\_parser()
85. options = parser.parse\_args()
86. func = options.func
87. **if** func == "server":
88. server()
89. **elif** func == "client":
90. word = options.word
91. client(word)


95. **if** \_\_name\_\_ == '\_\_main\_\_':
96. main()