CN_HW1

B04902017 李立譽

<Program Structure>

(i) Read file:

First, Open the config file through open() function

Second, Cut the path out of the string in config file.

```
file = open("config","r")
channel = file.read()
cha_len = len(channel)
channel = channel[6:cha_len-1]
```

(ii) Connect irc.freenode.net:

First, create socket and then set up hostname and port_number

```
s = socket.socket( socket.AF_INET, socket.SOCK_STREAM)
HostIP = 'irc.freenode.net'
Port = 6667
```

(iii) Join channel:

After connecting to the irc successfully, send “user”, “nick”, “join” commands respectively to join the channel.

```
s.connect( ( HostIP, Port) )
s.send(bytes("USER irobot17 irobot17 irobot17: irobot17\r\n" , "UTF-8"))
#print("##########################################\n")
s.send(bytes("NICK irobot17\r\n" , "UTF-8"))
#text = s.recv(2040)
#s.send(bytes('PONG ' + text.split() [1] + '\r\n'),"UTF-8")
#print("##########################################\n")
s.send( bytes("JOIN #%s\r\n" %channel, "UTF-8") )
```

(iv) Four Instructions:

Use a while loop to receive message until we shut down the program, and then respond based on the message we received

Ping:

Respond pong and the following message

@repeat:

cut the string after @repeat and then use socket.send() function send back

@convert:

if we receive a string which has “0x” at its beginning then

we convert it to decimal and send back.

similarly, if we receive a string which is normal number then we convert it to heximal(has “0x” at its beginning) and send back.

```
if text.find('@convert') != -1 :
    if text.find("0x") != -1:
        input = text[text.find('@convert') + 11:text_len]
        ans = int(input,16)
#       print(ans)
        s.send( bytes("PRIVMSG #" + channel + " : %d\r\n" %(ans), "UTF-8") )
    else:
        input = text[text.find('@convert')+9:text_len]
        input = int(input)
        ans = hex(input)
        #ans = ans[2:len(ans)]
#       print(ans)
        s.send( bytes("PRIVMSG #" + channel + " : "+ ans +"\r\n", "UTF-8") )
```

@ip:

receive a string of number and use three for loops to cut them down into four sections, and then check whether the result follow the several rules. Finally, we send back the answer.

```
if text.find('@ip') != -1:
    input1 = text[ text.find("@ip")+4 : text_len]
    i_len = len(input1) - 2     #because input1 has \r\n
    ip_num = 0
    ans = []
    for i in range(1,i_len+1):
        for j in range(i+1,i_len+1):
            for k in range(j+1,i_len+1):
                s1 = input1[0:i]
                s2 = input1[i:j]
                s3 = input1[j:k]
                s4 = input1[k:i_len]
                #print(s1 +" len = " + str(len(s1))+"\r\n")
                #print(s2 + " len = " +str(len(s2)) +"\r\n")
                #print(s3 + " len = " + str(len(s3)) +"\r\n")
                #print(s4 + " len = " + str(len(s4)) + "\r\n\r\n\r\n")
                if ( ((len(s1) > 1) and (s1[0] == '0')) or ((len(s2) > 1) and (s2
                    continue
                if len(s1) == 0 or len(s2) == 0 or len(s3) == 0 or len(s4) == 0:
                    continue

                if int(s1,10) > 255 or int(s2,10) > 255 or int(s3,10) > 255 or in
                    continue
                ip_num += 1
                full_str = s1+"."+s2+"."+s3+"."+s4
                ans.append(full_str)
    s.send( bytes("PRIVMSG #" + channel + " : " + str(ip_num) + "\r\n", "UTF-8")
    for i in range(0,ip_num):
        s.send( bytes("PRIVMSG #" + channel + " : " + ans[i] + "\r\n", "UTF-8") )
```

@help:

send all three functions above to host.

<Challenge>

This homework seems to be easy at first. However, while coding and implementing, I encounter some difficulties:

(i) Encoding & Decoding:
When receive and send message, we need to use some protocols to translate byte code.

(ii) Sending message form:
I forget to put ":" character into the sending message initially. Thus, I can't get the right answer in the chatbox.

(iii) Coding language:
Due to the fact that I haven't learned python before, I used C at first. However, because I found out that program written was more convenient to implement, I finally started learning python and finished the homework.

<Reflection>

I think this homework gives me a chance to be familiar with python and socket protocol. Moreover, the tutorial and spec describe the task precisely and also give me a clear concept about socket network. Those information really help and teach me a lot.