课程名称：Python语言与编程

实验名称：半双工的聊天程序

1. 实验目的

了解python简单的服务器与客户端通讯。

1. 实验内容

创建一个简单的，半双工的聊天程序。“半双工”的意思是当创建一个连接，服务启动的时候，只有一个人可以打字，另一个人只有在等到有消息通知他输入消息时，才能说话。一旦消息发送出去后，要等到有回复了才能发送下一条消息。一个人是服务端，另一个人是客户端。

1)半双工聊天,在本地记录通信双方的消息日志.(时间,通信内容等)

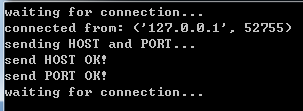
2)增加“名字服务器”，即客户端首先连接“名字服务器”，从名字服务器获得服务端的IP地址和端口，然后客户端建立与服务器的通讯连接，进行通讯。

1. 实验环境

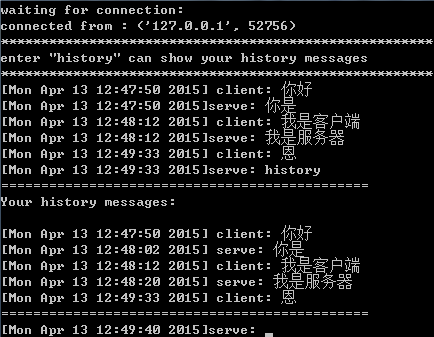
Python2.7.9

1. 实验结果

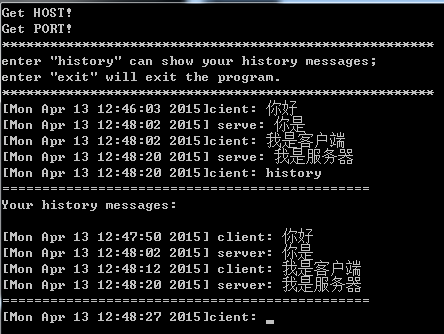
名字服务器：



服务器：



客户端：



1. 程序：

名字服务器：

from socket import \*

from time import ctime

HOST='localhost'

PORT=1008

BUFSIZ=1024

ADDR=(HOST,PORT)

tcpSerSock=socket(AF\_INET,SOCK\_STREAM)

tcpSerSock.bind(ADDR)

tcpSerSock.listen(5)

try:

while True://connection

print 'waiting for connection...'

tcpCliSock,addr=tcpSerSock.accept()

print 'connected from:',addr

print 'sending HOST and PORT...'

while True://send HOST

tcpCliSock.send('localhost')

data=tcpCliSock.recv(BUFSIZ)

if not data:

continue

else:

print 'send HOST OK!'

break

while True://send PORT

tcpCliSock.send('1005')

data=tcpCliSock.recv(BUFSIZ)

if not data:

continue

else:

print 'send PORT OK!'

break

finally:

tcpSerSock.close()

服务器：

from socket import \*

from time import ctime

HOST='localhost'

PORT=1005

BUFSIZ=1024

ADDR=(HOST,PORT)

tcpSerSock=socket(AF\_INET,SOCK\_STREAM)

tcpSerSock.bind(ADDR)

tcpSerSock.listen(5)

try:

while True://connection

print 'waiting for connection:'

tcpCliSock,addr=tcpSerSock.accept();

print 'connected from :', addr

print '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

print 'enter "history" can show your history messages'

print '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

while True://get client message

cdata = tcpCliSock.recv(BUFSIZ)

if not cdata:

continue

elif cdata=='exit':

break

else :

print '[%s] client: %s' % (ctime(), cdata)

handle=open('server.txt','a')

handle.write('[%s] client: %s\n' % (ctime(), cdata))

handle.close()

while True://send server message

sdata = raw\_input('[%s]serve: '%ctime())

if not sdata :

continue

elif sdata=='history':

print '=============================================='

print 'Your history messages:\n'

file\_object = open('server.txt','r')

for line in file\_object:

print line.strip('\n')

file\_object.close()

print '=============================================='

continue

else :

tcpCliSock.send(sdata)

handle=open('server.txt','a')

handle.write('[%s] serve: %s\n' % (ctime(), sdata))

handle.close()

break

finally:

tcpSerSock.close()

客户端：

from socket import \*

from time import ctime

HOST='localhost'

PORT=1008

BUFSIZ=1024

ADDR=(HOST,PORT)

tcpCliSock=socket(AF\_INET,SOCK\_STREAM)

tcpCliSock.connect(ADDR)

try:

while True:// Get server HOST

HOST2 = tcpCliSock.recv(BUFSIZ)

if not HOST2:

continue

else:

tcpCliSock.send('ok')

print 'Get HOST!'

break

while True:// Get server PORT!

PORT2 = int(tcpCliSock.recv(BUFSIZ))

if not PORT2:

continue

else:

tcpCliSock.send('ok')

print 'Get PORT!'

break

ADDR2=(HOST2,PORT2)//connertion

tcpCliSock=socket(AF\_INET,SOCK\_STREAM)

tcpCliSock.connect(ADDR2)

print '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

print 'enter "history" can show your history messages;'

print 'enter "exit" will exit the program.'

print '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

while True://send client message

cdata = raw\_input('[%s]cient: '%ctime())

if cdata=='exit':

tcpCliSock.send(cdata)

break

elif cdata=='history':

print '=============================================='

print 'Your history messages:'

file\_object = open('client.txt','r')

for line in file\_object:

print line.strip('\n')

file\_object.close()

print '=============================================='

continue

else:

tcpCliSock.send(cdata)

handle=open('client.txt','a')

handle.write('[%s] client: %s\n' % (ctime(), cdata))

handle.close()

while True://get server message

sdata = tcpCliSock.recv(BUFSIZ)

if not sdata:

continue

else:

print '[%s] serve: %s' % (ctime(), sdata)

handle=open('client.txt','a')

handle.write('[%s] server: %s\n' % (ctime(), sdata))

handle.close()

break

finally:

tcpCliSock.close()