

Computer Systems Administration

Semester 2 Python Assignment 2020

Student ID: A00268440

Course/Year: Computer Engineering 3

Assignment Title: Basic Encryption with Python

Brief Description

The purpose of this client-server application is to provide security of chat messages and save private information as an encrypted message.

Contents

Detailed Description3

Python Features4

User Manual.....5

References10

Detailed Description

This program aims to secure the messages between client and server.

On the server side, the process works by passing input messages to a defined hash function. And the hash function returns a hash string that keeps the encrypted information.

On the client side, both the encrypted message and the original message entered by the user are displayed on the screen. That means the user is allowed to see both messages.

Encryption secures online information and protects governments from countless daily attacks and is a vital component of the national defences.

What I tried to do is a basic form of a securing system using encryption.

Python Features

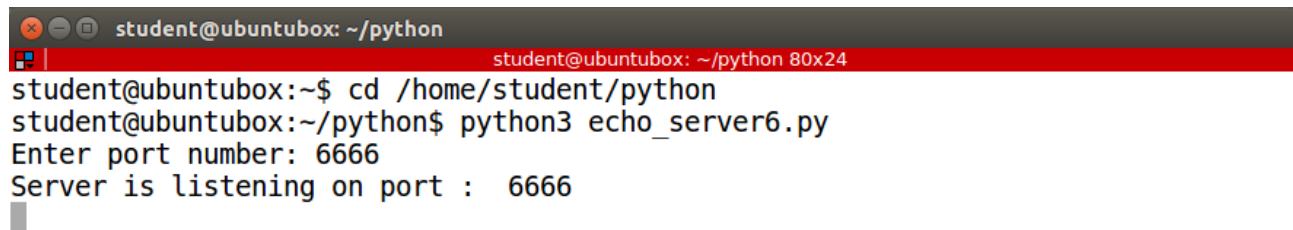
Features	Yes / No
The features from the 5 basic Echo exercises Command-line arguments (for server address/port number) Exception Handling (for client connecting/server binding) Transport Layer Security	Yes
Client GUI (EasyGUI only)	Yes
Functions and Threading Specify Function names and purpose Data Structures (Lists, Dictionaries/Shelves) Specify Data Structures names and purpose	Function Yes Threading No List Yes Dictionary Yes Shelves No
Additional command-line arguments Specify the purpose of each:	No
Additional exception handling Specify: Additional features in the client Specify: Additional features in the server Specify:	1- ValueError: When the user enters an inappropriate type of input 2- KeyboardInterrupt: When the user hits an interrupt key No Yes Hashing function to converting the input message into a encrypted message

User Manual

Server:

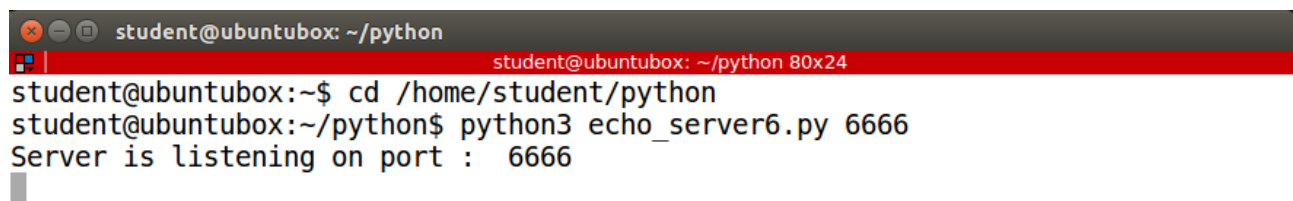
The server program can be run from the command-line with or without arguments:
If it runs without argument, it will prompt the user to enter a port number.

Without arguments:

A terminal window titled 'student@ubuntu: ~/python' with a red title bar. The terminal shows the following commands and output:

```
student@ubuntu:~$ cd /home/student/python
student@ubuntu:~/python$ python3 echo_server6.py
Enter port number: 6666
Server is listening on port : 6666
```

With argument specified port number:

A terminal window titled 'student@ubuntu: ~/python' with a red title bar. The terminal shows the following commands and output:

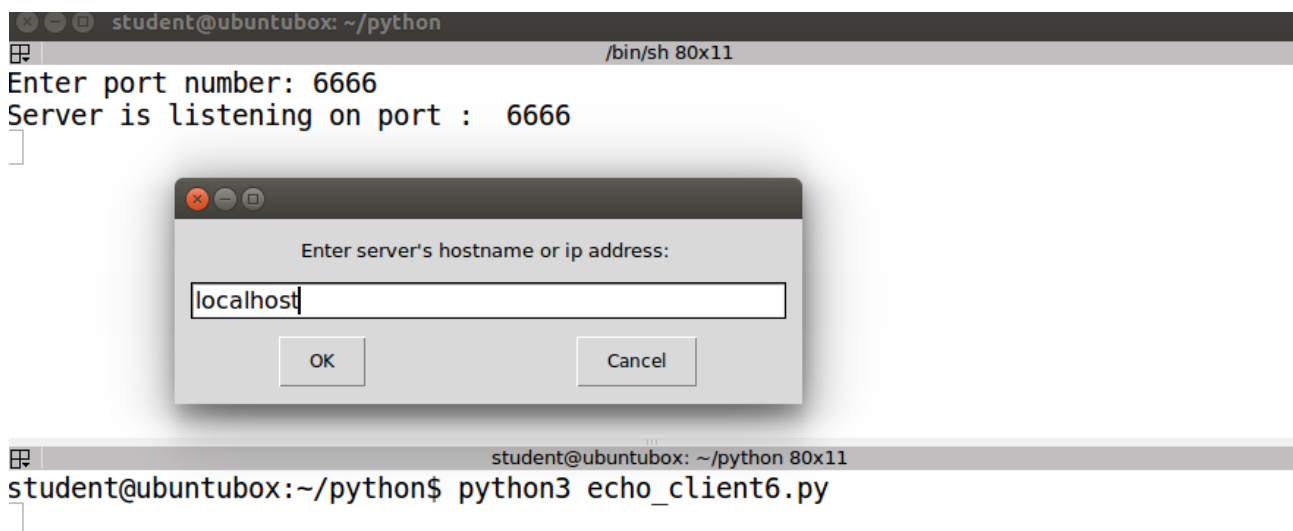
```
student@ubuntu:~$ cd /home/student/python
student@ubuntu:~/python$ python3 echo_server6.py 6666
Server is listening on port : 6666
```

Client:

The client program can be run from the command-line with and without arguments.
If run without arguments, it will prompt the user to enter the server ip address and the server port number.

Without any of the arguments:

First enter the hostname:

A terminal window titled 'student@ubuntu: ~/python' with a grey title bar. The terminal shows the following commands and output:

```
student@ubuntu:~$ python3 echo_client6.py
Enter port number: 6666
Server is listening on port : 6666
```

Below the terminal, a dialog box is shown with the title 'Enter server's hostname or ip address:'. It contains a text input field with 'localhost' and two buttons: 'OK' and 'Cancel'. Below the dialog box, the terminal window continues with the following command and output:

```
student@ubuntu:~/python$ python3 echo_client6.py
```

Then, enter the port number:

```
student@ubuntu: ~/python
/bin/sh 80x11
Enter port number: 6666
Server is listening on port : 6666
]

Enter the port number of the server:
6666
OK Cancel

student@ubuntu: ~/python 80x11
student@ubuntu:~/python$ python3 echo_client6.py
]
```

With only hostname argument:

Program will prompt you to enter the port number

```
student@ubuntu: ~/python
student@ubuntu: ~/python 80x11
student@ubuntu:~$ cd /home/student/python
student@ubuntu:~/python$ python3 echo_server6.py 6666
Server is listening on port : 6666
]

Enter the port number of the server:
OK Cancel

student@ubuntu: ~/python 80x11
student@ubuntu:~/python$ python3 echo_client6.py localhost
]
```

With arguments:

```
student@ubuntu: ~/python
student@ubuntu: ~/python 80x11
student@ubuntu:~$ cd /home/student/python
student@ubuntu:~/python$ python3 echo_server6.py 6666
Server is listening on port : 6666
Connection connection from: ('127.0.0.1', 55610)
]

Enter a message to send or enter q to quit:
OK Cancel

student@ubuntu: ~/python 80x11
student@ubuntu:~/python$ python3 echo_client6.py localhost 6666
Server: localhost on port 6666
```

If the user enters a wrong information, first error message will be displayed on the screen and then program will give you 3 options: exit, retry, change server details

```
student@ubuntu: ~/python
/bin/sh 80x11
Enter port number: 6666
Server is listening on port : 6666

[message box: Unable to connect, OK]

student@ubuntu: ~/python 80x11
student@ubuntu:~/python$ python3 echo_client6.py
student@ubuntu:~/python$ python3 echo_client6.py localhost 5
Server: localhost on port 5
```

```
student@ubuntu: ~/python
/bin/sh 80x11
Enter port number: 6666
Server is listening on port : 6666

[message box: connection error, quit, change server details, retry]

student@ubuntu: ~/python 80x11
student@ubuntu:~/python$ python3 echo_client6.py
student@ubuntu:~/python$ python3 echo_client6.py localhost 5
Server: localhost on port 5
```

If the user enters the correct information a messagebox will be displayed and user will be asked to enter a message:

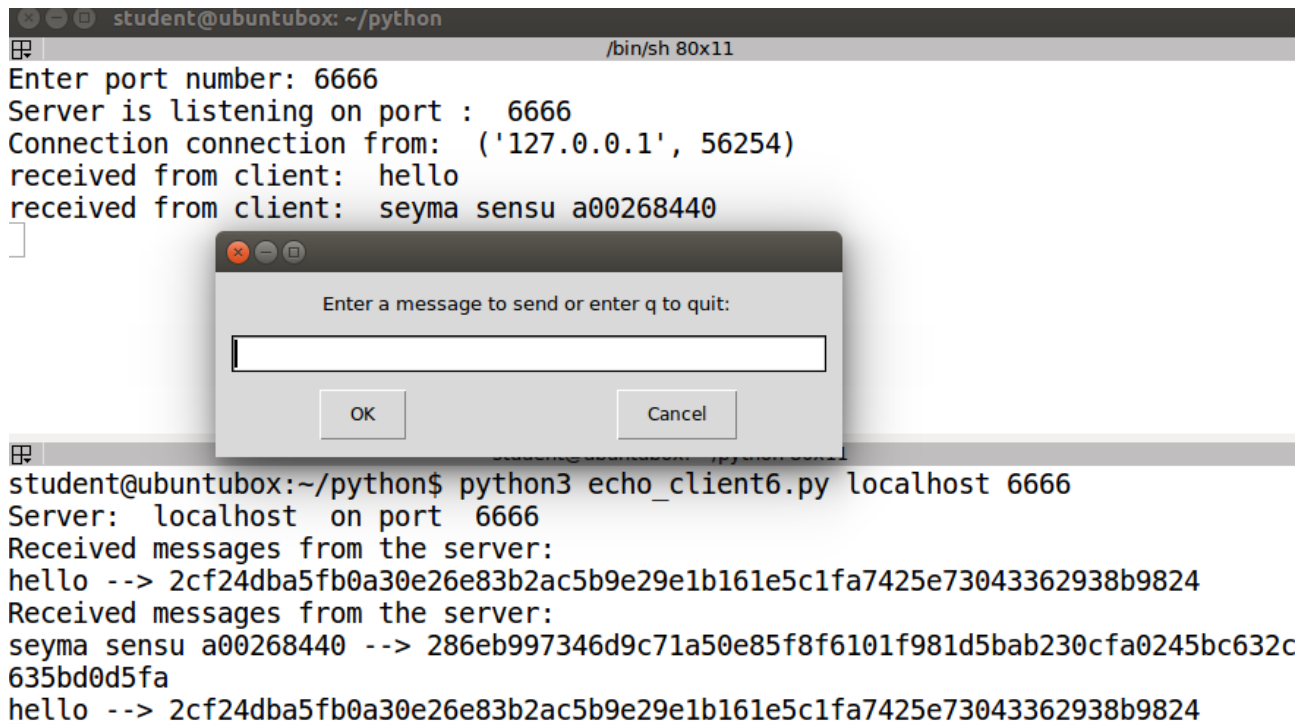
```
student@ubuntu: ~/python
/bin/sh 80x11
Enter port number: 6666
Server is listening on port : 6666
Connection connection from: ('127.0.0.1', 56254)

[message box: Enter a message to send or enter q to quit: OK, Cancel]

student@ubuntu:~/python$ python3 echo_client6.py localhost 6666
Server: localhost on port 6666
```

Messages entered by the user:

Program displays both the original message and hashed message.



The image shows a terminal window titled 'student@ubuntu: ~/python' with a shell prompt '/bin/sh 80x11'. The terminal output is as follows:

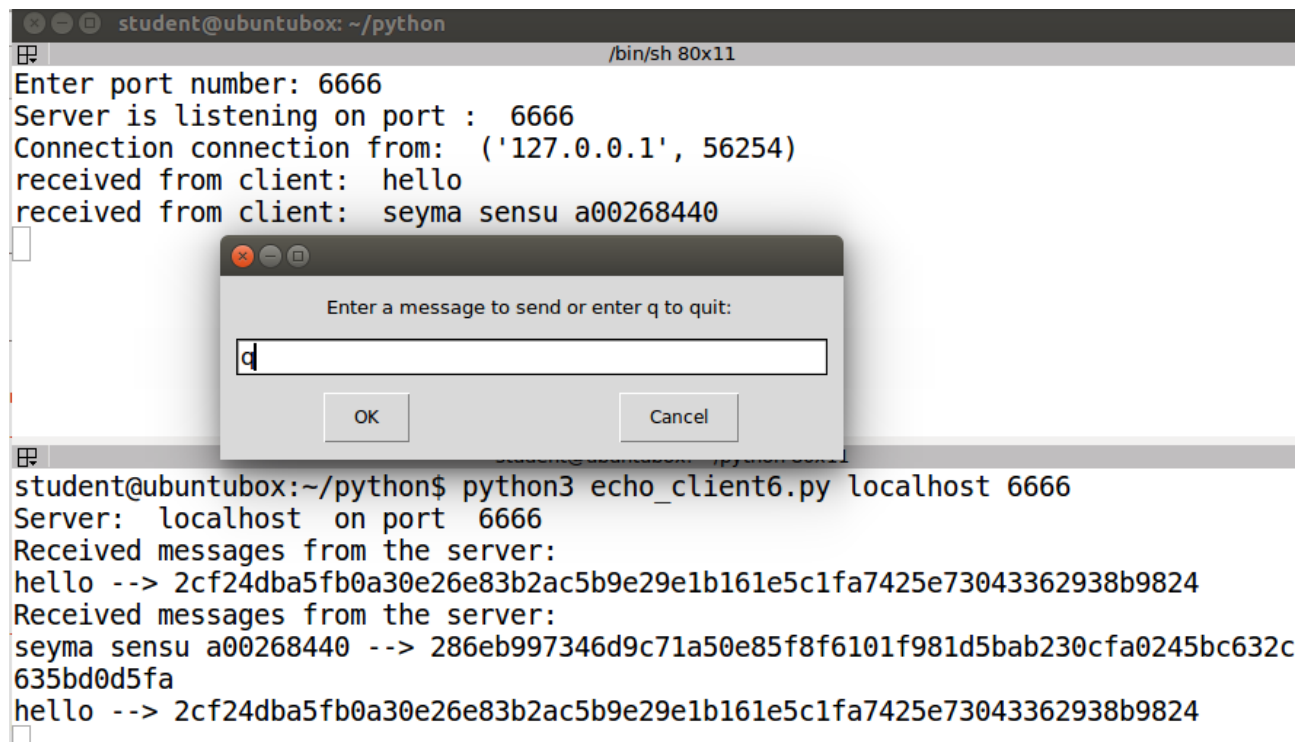
```
Enter port number: 6666
Server is listening on port : 6666
Connection connection from: ('127.0.0.1', 56254)
received from client: hello
received from client: seyma sensu a00268440
```

Overlaid on the terminal is a dialog box titled 'Enter a message to send or enter q to quit:'. It has a text input field and two buttons: 'OK' and 'Cancel'.

Below the dialog box, the terminal continues with the following commands and output:

```
student@ubuntu:~/python$ python3 echo_client6.py localhost 6666
Server: localhost on port 6666
Received messages from the server:
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Received messages from the server:
seyma sensu a00268440 --> 286eb997346d9c71a50e85f8f6101f981d5bab230cfa0245bc632c
635bd0d5fa
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
```

If the user enters q program will be terminated:



The image shows the same terminal window as above, but the dialog box now has the letter 'q' entered in the input field. The terminal output below the dialog box is identical to the previous block:

```
student@ubuntu:~/python$ python3 echo_client6.py localhost 6666
Server: localhost on port 6666
Received messages from the server:
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Received messages from the server:
seyma sensu a00268440 --> 286eb997346d9c71a50e85f8f6101f981d5bab230cfa0245bc632c
635bd0d5fa
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
```



```
student@ubuntu: ~/python
/bin/sh 80x11
Server is listening on port : 6666
Connection connection from: ('127.0.0.1', 56254)
received from client: hello
received from client: seyma sensu a00268440
received from client: q

-----
(program exited with code: 0)
Press return to continue
]

student@ubuntu: ~/python 80x11
student@ubuntu:~/python$ python3 echo_client6.py localhost 6666
Server: localhost on port 6666
Received messages from the server:
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
Received messages from the server:
seyma sensu a00268440 --> 286eb997346d9c71a50e85f8f6101f981d5bab230cfa0245bc632c
635bd0d5fa
hello --> 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
student@ubuntu:~/python$
```

References

<https://docs.python.org/3/library/hashlib.html>

Reference Sheets on Moodle