Computer Network
Project 1
Simple Multi-user TCP Chat
(200pts)

CSI4106-01

Fall, 2018

Prelim.

Before you do this homework, you must be fully aware of "**Project Policy Notice**"

Goal (you are expected to)

1. Learn a basic socket programming
   - TCP only
   - Sockets in Linux Environment
     - **Do not write a code in Windows.**
     - **The two OSs have different socket APIs.**
     - **That is, they are not compatible.**
2. Understand the "Client-Server" architecture
3. **Write a simple TCP-based multi-user chat application.**

Steps to do this project

1. Google or read your textbook to follow up **the Client code** we provide.
   - Check the next slide.
2. Understand what **select()** does exactly
   - (hint) sys.stdin *(standard input stream)* can be considered as a socket in linux
3. **Write your own codes for multi-user chatting application running in real-time.**

Objectives

- You must complete **`cli.py`** we provide.
- You must write your own code **`srv.py`** we provide.
- The two codes must take the two parameters
  - IP address and Port number
  - e.g.) `python srv.py 0.0.0.0 5000`
  - e.g.) `python cli.py 127.0.0.1 5000`

What are these functions?

- **socket**(socket.AF_INET, socket.SOCK_STREAM)
- **setsockopt**(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
- **select**([], [], [])
- **bind**(("0.0.0.0", 7777))
- **listen**(5)
- **connect**((host, port))
- **accept**()
- **close**()

`cli.py` (note this is incomplete)

```
import socket, sys, select
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((host, port))
while True:
  sr, sw, se = select.select([sys.stdin, sock], [], [])
  for s in sr:
    if s == sock:
      data = s.recv(1024)
      print(data)
    else:
      sock.send(input())
```

Guidelines – Server

```
[root@localhost p1]# python srv.py 127.0.0.1 8888
Chat Server started on port 8888.
> New user 127.0.0.1:31335 entered (1 user online)
[127.0.0.1:31335] Hello

> New user 127.0.0.1:31341 entered (2 users online)
[127.0.0.1:31341] World

< The user 127.0.0.1:31341 left (1 user online)
< The user 127.0.0.1:31335 left (0 user online)

KeyboardInterrupt
[root@localhost p1]#
```

Guidelines – Client 1

```
[root@localhost p1]# python cli.py 127.0.0.1 8888
> Connected to the chat server (1 user online)
[You] Hello
> New user 127.0.0.1:31341 entered (2 users online)
[127.0.0.1:31341] World

< The user 127.0.0.1:31341 left (1 user online)

KeyboardInterrupt
[root@localhost p1]#
```

Guidelines – Client 2

```
[root@localhost p1]# python cli.py 127.0.0.1 8888
> Connected to the chat server (1 user online)
[You] World
[You]
< You have been disconnected.

[root@localhost p1]#
```

Your report must include

- **Introduction/Reference** (5pts)
  - Language you used and so on.
- **Flow chart or Diagram** (20pts)
  - Must show the logic of your program
  - Focus on describing how your client and server work.
- **At least 3 snapshots** which prove your codes are working well. (5pts)

Your report must include (cont'd)

- **Logical explanations block by block in detail**. (40pts)
  - It is different with brief comments in your source code!!!
  - In your report, write what the blocks do and why you implemented those functions.
- **All explanations of the 8 functions** in "What are these functions?" slide. (10pts)
- **What is the function of `select()`?** (20pts)
  - Explanation + Pros and Cons

We will test your code as follows.

- OS: Ubuntu Linux
- Language: Python2 / Python3 / C language
- Your codes can
  - Run with **custom IP and Port** (15pts)
  - Work perfectly during evaluation **with no error** (30pts)
  - Be **terminated** by Ctrl+C or Ctrl+Z (15pts)
  - **Close the sockets** (We will check this by **`netstat`**) (40pts)

You will get 0 points if you…

- Copy your friend's codes
  + Change a little bit of them.
  + Wish that TAs don't catch that.

- Use a 3rd-party API or codes.

- Implement the server with "Multi-thread"

Deliverable

- **Only one zip file of "YourID_p1.zip"**
  - **If your ID is 2018147123, 2018147123_p1.zip should be your deliverable file name.**

- **In the zip file only the three files must be included without any folder**
  - **report.pdf**
  - **cli.py or cli.c**
  - **srv.py or srv.c**
    - **if you use C language, include compile.sh as well**

- **DUE DATE**

  18/Oct/2018 23:59:59 KST

  No exception for exceeding deadline

- **Delay Policy**

  -33%pts for ~19/Oct 23:59:59

  -66%pts for ~20/Oct 23:59:59

  -100%pts for 21/Oct 00:00:00~

Score Policy: *Max. 200pts*

1	Not submitted / not working / missing files	**0 pts**
2	Overdue ➔ Delay	**-33% pts/day**
3	**The rules or directions whose scores are not specified are not followed**	-10 pts/rule
4	**Any 3rd party framework is used**	**0 pts**
5	**Plagiarizing / Over-implementation (Any kinds of Suspicion of Code-copy)**	**0 pts**
6	**Impolite Report / Lack of Comments**	**0 pts / -50% pts**