EE323 Project 1

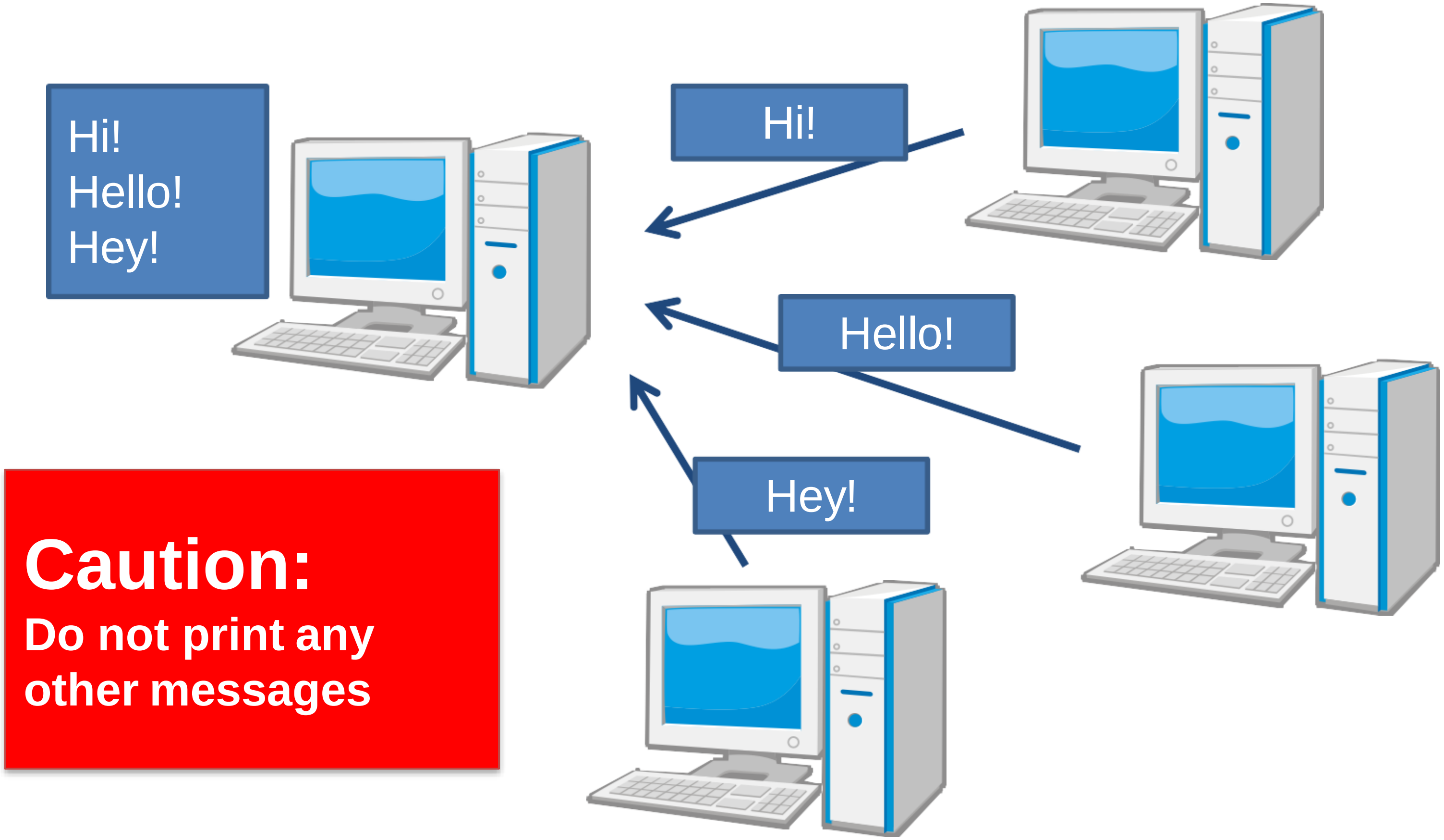
Introduction to Socket Programming

3/24 2020

TA Daeyang Cho

Scenario

- One server receives messages from multiple clients
- The server prints these messages on standard output



Caution:

Do not print any other messages

Requirements: Server

- Should be listening for messages to a port known to the clients
- Should be able to receive messages from multiple clients
  - Each message should be separated by **ENTER**
  - No need to be an event-driven or multi-threaded server
  - Use fork() to make a child process for each client
- Should take “-p” and the port number to listen on as arguments
- If the server cannot bind to the port that you specify, a message should be printed on standard error and the program should exit
- You shouldn't assume that your server will be running on a particular IP address, or that clients will be coming from a predetermined IP a ddress

Requirements: Client

- Should receive the message from standard input

- The client should send the message when you hit ENTER
- Hitting >ENTER< twice is an instruction for the client to exit
  - A single ENTER should not be transmitted

- Should take "-p" and the server's port number, and "-h" and the host that the server is running in any order

- Both the client and server should generate an appropriate error message and terminate when given invalid arguments

Pipe & Redirection

- Pipe
  - Get standard input of one process from standard output of another process
  - ex) $ ls –al | grep server

- Redirection
  - Get standard input from a file or write standard output to a file
  - ex) $ client –p 1234 –h 123.123.123.123 < input.txt

    $ server –p 1234 > output.txt

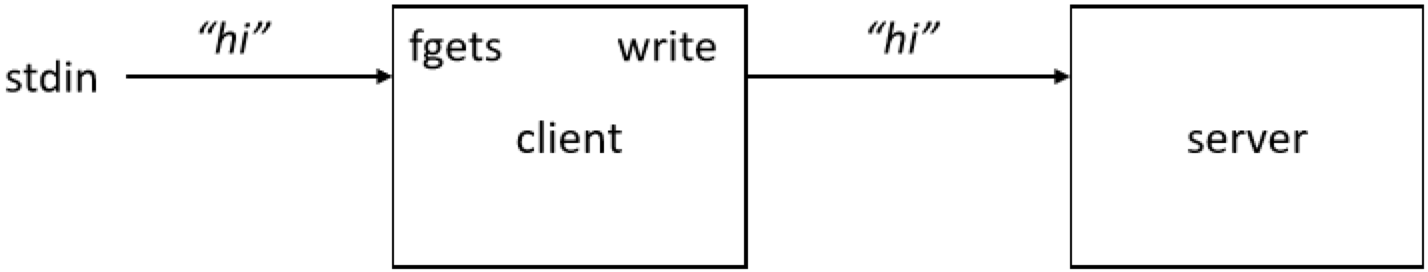
Caution:

This is NOT an ARGUMENT

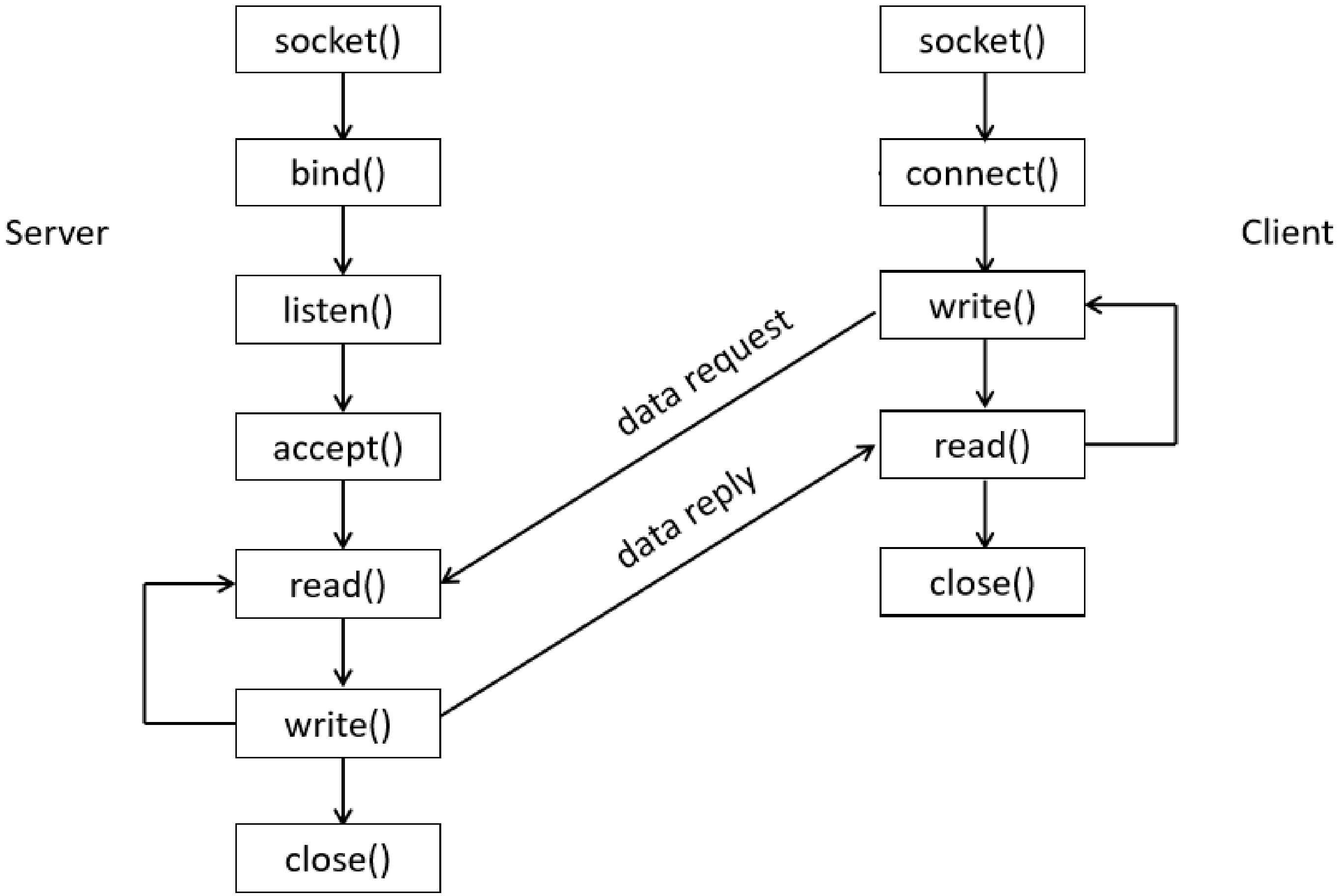
TCP echo client/server

- The client reads a line of text from its ***standard input*** and writes the line to the server

- The server reads the line from its network input and prints it as ***standard output***



Client – Server socket API



Suggestion

- Read uploaded Beej's guide to network programming carefully before start the assignment
  - Fully understand the example stream server and client code line-by-line

- Start from input argument parsing
  - Hint: getopt()

- Setup TCP socket server and client using the parsed input argument
  - Try to say "hello" from server to client, and client to server

- How to get input? How to detect the control sequence? How to send the data? How to receive the data? How to print it?
  - Hint: try with several I/O functions

Makefile

- Compile
  - $ gcc server.c –o server
  - $ gcc client.c –o client

- Write a “Makefile” and use it for compile
  - $ make

- Sample “Makefile”

```
all: server client

server:
            gcc –o server server.c

client:
            gcc –o client client.c

clean:
            rm server client
```

Makefile

- **If your Makefile doesn't work (can't compile your code), you cannot get a point!!**

- Test & compile will be processed on **eelab5** and **eelab6** in Haedong lounge
- Thus, if you are working on your own machine, don't forget to test it on these two machines before submission

```
all: server client

server:
            gcc –o server server.c

client:
            gcc –o client client.c

clean:
            rm server client
```

Submission

- Due: **4/6 (Mon.) 23:59:00**

- One tar file which contains “readme”, “server.c”, “client.c”, and “Makefile”

- Use KLMS to submit your assignments
  - Please check KLMS submission links before deadline
  - If you have any troubles to access it, please let TAs know.

- Your submission should be one gzipped tar file whose name is YourStudentID_assign#.tar.gz ex)2020xxxx_assign1.tar.gz

- How to make the gzipped tar file
  - $ tar cvzf 2020xxxx_assign1.tar.gz readme server.c client.c Makefile

Late Policy

- Late penalty
  - 10% late penalty per day
  - Can’t submit after 48 hours is elapsed from due date.
- Example
  - After 16 hours -> your grade = original point * 0.9
  - After 1day & 3hours -> your grade = original point * 0.8
  - After 2days & 1 second -> your grade = 0
- Token
  - Every student will get 3 tokens for deadline extension on this semester
  - One token can offset one day delay
  - If you submitted your assignment later than the deadline, your token automatically used for the delay

Others

- **Do NOT copy and paste** someone else’s code including publicly available source code
- We consider such cheating very seriously

- **Please read assignment document in KLMS carefully**

- Start the assignment as quickly as possible

- Any questions?