IPv6 Socket Programming

2019-11-28

# Object

IPv6 is an emerging protocol which will replace the IPv4 protocol and will become the most popular protocol within 7 years. Hence, it is important for us to understand this important protocol. The Socket interface is API to bridge between applications and the kernel. The socket programming is useful when an application deals with communicating another process in the same machine or different machines. The socket programming in the IPv6 network is proficient to students.

# Communication protocol

Students must write socket programming in the client and the server. The student’s client must connect to the designated server running in 2001:0:53aa:64c:49:69d3:8c6e:5548 IPv6 address and 50000 TCP port number. The designated server uses the following protocol to communicate with the client.

**\* If you running your program in school network, connection may blocked by firewall**

1. First Stage

After connecting to the server the designated server will ask your ID and your server’s IP address and port number. Your server should be running at this time and have a valid IPv4 address and TCP port number.

1. Second stage

The designated starts the client on its machine and opens the second connection to your server on the given the IP address and the port number. The new client will deliver the 20-byte string (token) on the second connection. Your server must pass the token to your client and your client must return this token to the designated server on the first connection.

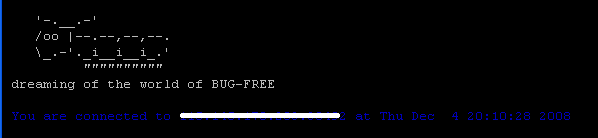
The following figure illustrate the protocol.



Figure Protocol

# Instruction

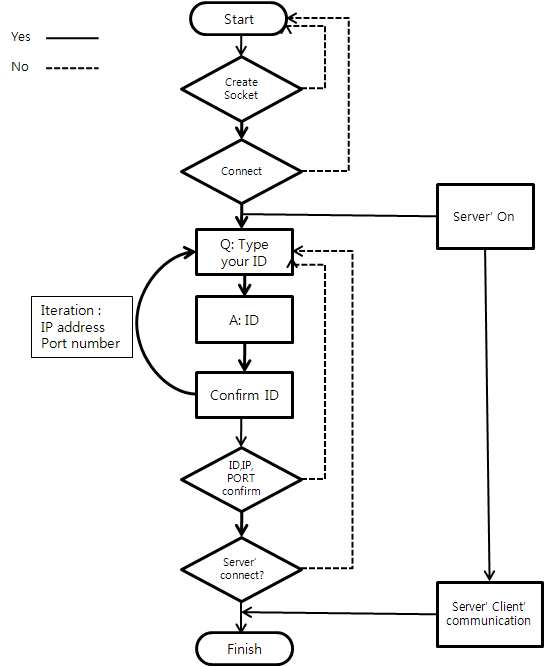
1. In the beginning the banner will welcome you to my machine.



1. You must return the token received from the second connection to the first connection.
2. All the input must be terminated by the carriage return.
3. You can also connect to the designated server using “ping” for the connectivity test, nothing else.
4. Report immediately if you find a bug.
5. Make sure that I am available to connect your server.

# Program details

The following figure shows the flow of the program.



# Concurrent Programming (optional)

If you want a higher score, you can try to make your server running concurrently.

There is an option for students who want to challenge themselves.

Our client sends multiple requests from several process at the same time.

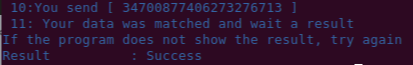
You should connect all of client process and return the token of **all.**

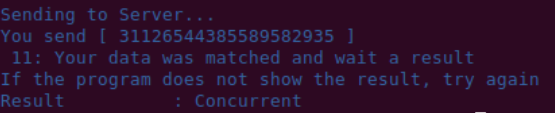
Our server will accept the token only if all tokens match exactly.

If your program shows the “Result : Concurrent” your program is works fine.

If your program does not show the any result just try again, it may had a problem in synchronization of our IPv6 server.

Example)

 **Normal Success**

 **Concurrent Success**

# Deliverables

## Upload source code

Upload your program onto icampus.

Do not even think about copying other’s code.

## Upload your report

Complete the report in MS Word and upload onto “iCampus”. In the report you must include all of following. Name your file as your\_Korean\_name.doc. Do not zip your file.

### Screenshot of successful FTP session

### 20-byte string that you have received from the server

### Your unique experience

### Your setup to connect to the Ipv6 server

### Screenshot of the successful conversation