

## Assignment #1

## Client-Server Comp&amp;Web Tech

ให้ นศ. ออกแบบ โปรแกรมประยุกต์อย่างง่าย เพื่อใช้ประโยชน์จาก Client-Server Computing โดยมีรูปแบบดังนี้

- Server ทำการรอให้บริการอยู่ที่ TCP พอร์ต 5000
- Client เมื่อทำการเชื่อมต่อกับ Server จะต้องแนะนำตนเองก่อน โดยมีรูปแบบของข้อความคือ
- HI, I am ABC
- Server จะตอบกลับด้วยข้อความ READY หาก Client ส่งข้อความแนะนำมาถูกต้อง แต่จะส่งข้อความ SORRY หาก Server ไม่สามารถเข้าใจข้อความที่ส่งมาได้ และ Server ตัดการติดต่อ
- Client จะส่งคำร้องขอ เพื่อให้ Server ประมวลผล (นศ. ออกแบบเองว่า ต้องการให้ Server บริการ ประมวลผลเกี่ยวกับอะไร)
- Server แจ้งผลลัพธ์กลับไป พร้อมแสดงผลบนหน้าจอ ว่า ได้ส่งผลลัพธ์อะไร ไปยัง Client ชื่ออะไร (ตามที่ได้แจ้งมาในขั้นตอนก่อนหน้านี้)
- Client ส่งข้อความ BYE และ Server ตัดการติดต่อ

assignServer.js

```

1  var net = require('net');
2
3  var HOST = '127.0.0.1';
4  var PORT = 5000;
5
6  var server = net.createServer(function (socket){
7
8      var appState = 0 ;
9      var clientName ;
10
11      socket.on('data',function(data){
12
13          var str = (" " + data).trim();
14
15          switch(appState){
16
17              case 0:
18                  if(str.indexOf('Hi, i am ')==0){
19
20                      clientName = str.substring('Hi, i am '.length);
21                      console.log("client name is " + clientName);
22                      socket.write("READY");
23                      console.log("READY");
24                      appState = 1 ;
25                  }
26                  else{
27                      socket.write("SORRY");
28                      console.log("SORRY");
29                      socket.destroy();
30                  }
31                  break;

```

```

32
33
34     case 1:
35         //process data
36         //format input from client
37         //cleint.write('55+35+512+117');
38         var temp = parseInt(str.substring(0,2)) +
39                     parseInt(str.substring(3,5)) +
40                     parseInt(str.substring(6,9)) +
41                     parseInt(str.substring(10,13)) ;
42
43         temp = temp.toString();
44         console.log('data: ' + temp);
45         socket.write(temp);
46
47         break;
48     }
49 });
50
51 server.listen(PORT, HOST);
52 console.log('Server listening on ' + HOST + ':' + PORT);

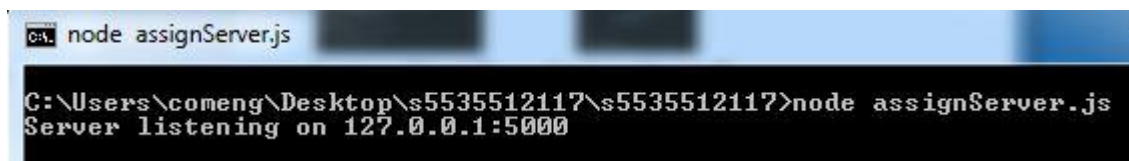
```

### อธิบายการทำงานของโปรแกรม

กำหนดตัวแปร HOST และ PORT ที่เก็บค่า IP loopback (127.0.0.1) และ Port 5000 ตามลำดับ จากนั้นกำหนดตัวแปร server เพื่อสร้าง socket ฝั่ง server ไว้ใช้งาน โดยภายในได้สร้างตัวแปรขึ้นมา 2 ตัว ได้แก่ appState มีไว้เก็บค่าสถานะการทำงานของ Server และ clientName ไว้เก็บค่าที่ Client ส่งมาและ server ทำการตรวจสอบว่าข้อมูลตรงกันหรือไม่

ซึ่ง appState โดยกำหนดค่าเริ่มต้นเท่ากับ 0 มีไว้เพื่อเปลี่ยนสถานะการทำงานภายในฝั่ง Server ให้มี 2 ลักษณะการทำงานคือ 1.appState = 0 คือการตรวจสอบข้อความของ Client ว่าถูกต้องหรือไม่ หากถูกต้องให้ส่งข้อความกลับไป Client ว่า READY และเปลี่ยน appState = 1 เพื่อรอ Client Request แต่หากไม่ใช่ให้ส่งข้อความ SORRY ไปแทนและทำการตัดการเชื่อมต่อ 2.appState = 1 คือกระบวนการหลังจากตรวจสอบเรียบร้อยแล้ว จะเป็นการ Process Data ตาม Client Request ที่ส่งข้อมูลมา ซึ่งจากโค้ดโปรแกรมเป็นการส่งรหัสนักศึกษาใน Format (xx+xx+xxx+xxx) แล้วจึงนำชุดตัวเลขทั้งหมด 4 ชุดมาบวกเข้าด้วยกัน แล้วส่งผลลัพธ์กลับไปยัง Client

### ผลลัพธ์การทำงานของโปรแกรม



```

C:\> node assignServer.js
C:\Users\comeng\Desktop\s5535512117\s5535512117> node assignServer.js
Server listening on 127.0.0.1:5000

```

เมื่อ run ไฟล์ assignServer.js เป็นการแสดง IP & Port ที่ Server ทำงานอยู่

```

C:\> node assignServer.js

C:\Users\comeng\Desktop\s5535512117\s5535512117>node assignServer.js
Server listening on 127.0.0.1:5000
client name is Jane
READY
data: 719

```

ผลลัพธ์หลังจากมี Client ชื่อ Jane ทำการติดต่อมา

- แสดงข้อความ READY เพื่อบอกว่า Server พร้อมรับ Client Request
- แสดงผลลัพธ์ Process Data ที่ Client ส่งมาให้

assignClient.js

```

assignClient.js
1  var net = require('net');
2
3  var HOST = '127.0.0.1';
4  var PORT = 5000 ;
5
6  var client = new net.Socket();
7
8  client.connect(PORT,HOST,function(){
9
10     console.log('Connected');
11     client.write('Hi, i am Jane');
12     //client.write('s5535512117');
13 });
14
15 client.on('data', function(data){
16
17     var resp = (" " + data).trim();
18
19     if(resp == "READY"){
20
21         console.log('55+35+512+117 = ?');
22         client.write('55+35+512+117');
23
24     }
25     else {
26
27         console.log('My Result = ' + data);
28         client.destroy();
29     }
30 });
31
32 client.on('close', function(){
33
34     console.log('BYE');
35 });

```

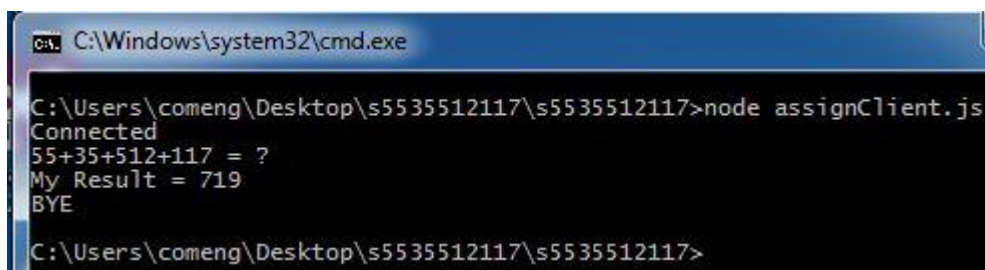
### อธิบายการทำงานของโปรแกรม

กำหนดตัวแปร HOST และ PORT เช่นเดียวกับโปรแกรม assignServer.js เพื่อให้ง่ายต่อการเรียกใช้และง่ายต่อการทำความเข้าใจ ต่อจากนั้นกำหนดตัวแปร client เพื่อทำการสร้าง socket เพื่อใช้งานในฝั่ง client แสดงข้อความว่า “Connected” แล้วส่งข้อมูลไปยัง Server ไปทำการตรวจสอบเพื่อพร้อมในการใช้งาน

Event เมื่อมีข้อมูลส่งมาจาก Server ให้เก็บข้อมูลนั้น ไว้ที่ data แล้วประกาศตัวแปร resp โดยทำการลดช่องว่างด้านหน้าและด้านหลังตัวแปรที่ไม่มีประโยชน์จากตัวแปร data ออกให้หมด ต่อมาตรวจสอบเงื่อนไขว่าข้อมูลนั้นเป็นค่า READY ที่ Server บอกให้ Client ทราบว่าพร้อมในการ Process Data แล้ว ให้ Client Request ข้อมูลไปที่ Server ได้โดยทันที

เมื่อการทำงานของ Server เสร็จเรียบร้อยแล้ว ก็จะส่งผลลัพธ์การประมวลผลมาที่ฝั่ง Client เนื่องจาก javascript มีพฤติกรรมการทำงานในเชิงคู่ขนาน ทำให้เลือกค่าที่ส่งมานอกเหนือจากข้อมูลก่อนหน้านี้คือ READY เพื่อให้ Client แสดงผลลัพธ์ที่ได้มาจาก Server สุดท้ายจึงตัดการเชื่อมต่อ และ Event.close จึงทำงานแล้วแสดงข้อความว่า BYE จบการทำงานของโปรแกรม

### ผลลัพธ์การทำงานของโปรแกรม



```

C:\Windows\system32\cmd.exe
C:\Users\comeng\Desktop\s5535512117\s5535512117>node assignClient.js
Connected
55+35+512+117 = ?
My Result = 719
BYE
C:\Users\comeng\Desktop\s5535512117\s5535512117>

```

เมื่อ run ไฟล์ assignClient.js