

Messenger

41088510 劉嘉蕎

A. Server 端

1. 按下 activate 按鈕時，抓取 IP Address，並且啟動服務執行緒

```
private void button1_Click_1(object sender, EventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    IPAddress[] ip = Dns.GetHostEntry(Dns.GetHostName()).AddressList;
    foreach (IPAddress it in ip)
    {
        if (it.AddressFamily == AddressFamily.InterNetwork)
            button1.Text = it.ToString();
        else button1.Text = "";
    }

    Th_Svr = new Thread(ServiceOn);
    Th_Svr.IsBackground = true;
    Th_Svr.Start();
    button1.Enabled = false;
}
```

2. 每當伺服器端接受新的用戶，就為他建立新的監聽執行緒

```
private void ServiceOn()
{
    Server = new TcpListener(IPAddress.Parse(button1.Text), 2021);
    Server.Start();
    while (true)
    {
        try
        {
            Client = Server.AcceptSocket();
            Th_Clt = new Thread(Listen);
            Th_Clt.IsBackground = true;
            Th_Clt.Start();
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }
}
```

3. 監聽執行緒會先試著接收封包，再進行解碼

```
private void Listen()
{
    Socket Sck = Client;
    Thread Th = Th_Clt;
    while (true)
    {
        try
        {
            byte[] B = new byte[Sck.ReceiveBufferSize];
            int len = Sck.Receive(B);
            Console.WriteLine("Receive B ");
        }
    }
}
```

4. 解碼的邏輯

- i. 轉換成 string 之後第一個字元指示出封包的意涵，之後的字元 str 代表內容

```
string Msg = Encoding.Default.GetString(B, 0, len);  
string flag = Msg.Substring(0, 1);  
string str = Msg.Substring(1);
```

- ii. 第一個字元為 0，表示需要建立名為 str 的新用戶，更新用戶名單與 hash table

```
if (flag == "0")//建立新用戶  
{  
    HT.Add(str, Sck);  
    listBox1.Items.Add(str);  
}
```

- iii. 第一個字元為 1，則先廣播傳送者的名字，再廣播傳送訊息 str

```
foreach (string n in HT.Keys)  
{  
    if (HT[n] == Sck) name = n;  
}  
  
if (flag == "1")//接收+傳送訊息  
{  
    foreach (string n in HT.Keys)  
    {  
        SendTo("0" + name, n);  
        SendTo("1" + str, n);  
        Console.WriteLine(name + ':' + str);  
    }  
}
```

- iv. 第一個字元為 8，表示用戶先通知即將傳送檔案和其附檔名，接下來會立即傳送完整檔案的封包；伺服器則是先廣播通知即將傳送檔案和其附檔名，接下來廣播傳送完整檔案封包

```
if (data_flag == true)  
{  
    foreach (string n in HT.Keys)  
    {  
        Socket s = (Socket)HT[n];  
        s.Send(B, 0, B.Length, SocketFlags.None);  
        Console.WriteLine("Sever has send a file.");  
    }  
    data_flag = false;  
}  
  
if (flag == "8")//副檔名  
{  
    foreach (string n in HT.Keys)  
    {  
        SendTo("8" + str, n);  
    }  
    data_flag = true;  
    Console.WriteLine("data_flag has been set");  
}
```

- v. 第一個字元為 9，表示刪除傳送者，更新用戶清單、hash table 並結束該用戶的專屬監聽執行緒

```
if (flag == "9")//刪除用戶
{
    listBox1.Items.Remove(name);
    HT.Remove(name);
    Th.Abort();
}
```

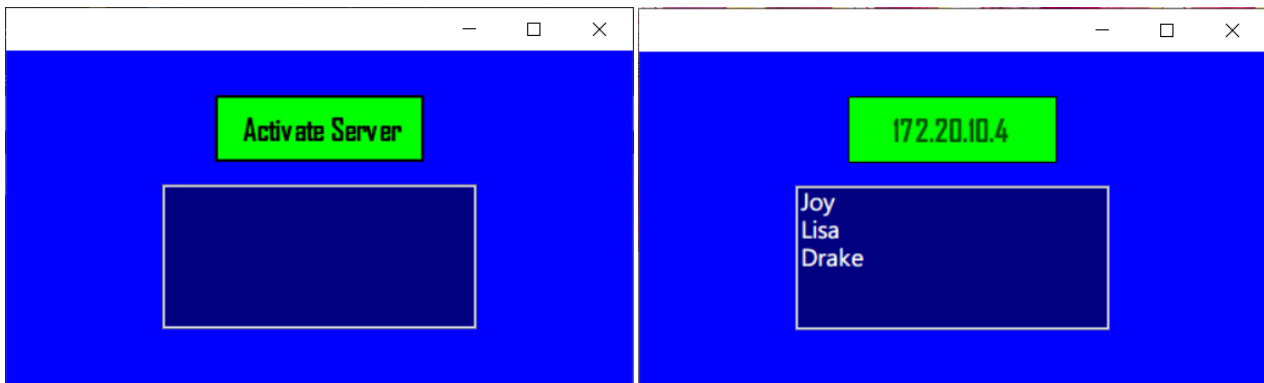
5. 傳送訊息的函式

```
private void SendTo(string Str, string user)
{
    byte[] B = Encoding.Default.GetBytes(Str);
    Socket Sck = (Socket)HT[user];
    Sck.Send(B, 0, B.Length, SocketFlags.None);
}
```

6. 關閉表單時，結束所有執行緒並停止服務

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.ExitThread();
    if(Server != null) Server.Stop();
}
```

7. 伺服器端初始畫面、啟動並接受用戶的畫面



B. Client 端

1. 建立表單時，調整表單控制項的細項(字體、對齊等)，在此不羅列
2. 用戶輸入名字並按下開始，即嘗試連線、開始監聽執行緒、發送自己的名字並載入聊天介面
3. `init_chat()` 為表單控制項的調整，在此不羅列

```
private void button1_Click(object sender, EventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    if (textBox2.Text == "" || textBox2.ForeColor == Color.DimGray) return;
    try
    {
        IPEndPoint EP = new IPEndPoint(IPAddress.Parse(ip), port);
        s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        s.Connect(EP);
        Th = new Thread(Listen);
        Th.IsBackground = true;
        Th.Start();

        Send("0" + textBox2.Text);
        myname = textBox2.Text;
        flowLayoutPanel0.Visible = false;
        flowLayoutPanel0.Enabled = false;
        init_chat();
    }
}
```

4. 監聽執行緒會先試著接收封包，再進行解碼

```
private void Listen()
{
    EndPoint ServerEP = (EndPoint)s.RemoteEndPoint;
    int len = 0;
    string Msg;
    string flag;
    string str;
    while (true)
    {
        try
        {
            byte[] B = new byte[s.ReceiveBufferSize];
            len = s.ReceiveFrom(B, ref ServerEP);

```

5. 解碼的邏輯

- i. 轉換成 string 之後第一個字元指示出封包的意涵，之後的字元 str 代表內容

```
Msg = Encoding.Default.GetString(B, 0, len);
flag = Msg.Substring(0, 1);
str = Msg.Substring(1);
```

- ii. 接收文字訊息:先收到傳送者的名字，再收到文字訊息

```
if (flag == "0")//對方名字
{
    name = str;
}
if (flag == "1")//對方訊息
{
    name_[count] = name;
    content_[count] = str;
    Console.WriteLine(name_[count] + ": " + content_[count]);
    count++;
}
```

- iii. 接收檔案封包:先被通知即將收到檔案與副檔名，再將檔案儲存到指定路徑(檔名是根據檔案內容產生的一串數字)

```
if (data_flag == true)
{
    string filename = null;
    foreach (char c in str.Substring(20, 25))
    {
        filename += Convert.ToInt32(c);
    }
    File.WriteAllBytes("D:\\joyli\\Downloads\\Lisa\\" + filename + extension, B);
    data_flag = false;
}

if (flag == "8")//副檔名
{
    data_flag = true;
    extension = str;
}
```

6. 傳送訊息的程式

7. 傳送訊息的按鈕:無文字時不能傳送、傳送後完清空文字框

```
private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "") return;
    Send("1" + textBox1.Text);
    textBox1.Text = "";

    byte[] B = Encoding.Default.GetBytes(Str);
    s.Send(B, 0, B.Length, SocketFlags.None);
}
```

8. 傳送檔案的按鈕:跳出檔案總管供用戶選擇檔案，而後傳送副檔名與檔案(檔案流向:fileStream>>memoryStream>>byte[] array)

```
private void button3_Click(object sender, EventArgs e)
{
    var fileContent = string.Empty;
    var filePath = string.Empty;

    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.InitialDirectory = "C:\\Users\\joyli\\OneDrive\\圖片";
        openFileDialog.Filter = "All files (*.*)|*.*";
        openFileDialog.FilterIndex = 2;
        openFileDialog.RestoreDirectory = true;

        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            filePath = openFileDialog.FileName;
            var fileStream = openFileDialog.OpenFile();
            MemoryStream ms = new MemoryStream();
            fileStream.CopyTo(ms);

            byte[] file = ms.ToArray();
            //Console.WriteLine(file.Length);
            Send("8" + Path.GetExtension(filePath));
            s.Send(file, 0, file.Length, SocketFlags.None);
            Console.WriteLine("file has sent");
            ms.Close();
            fileStream.Close();
        }
    }
}
```

9. 每 20 毫秒更新一次對話內容，調整表單控制項故在此不說明

```
private void timer1_Tick(object sender, EventArgs e)
{
    for (int i = have_shown; i < count; i++)
    {
        l1[i].Text = name_[i] + ":";
        l2[i].Text = content_[i];
        p[i].Location = new Point(3, (Int32)height);
        height += p[i].Height;
        have_shown++;

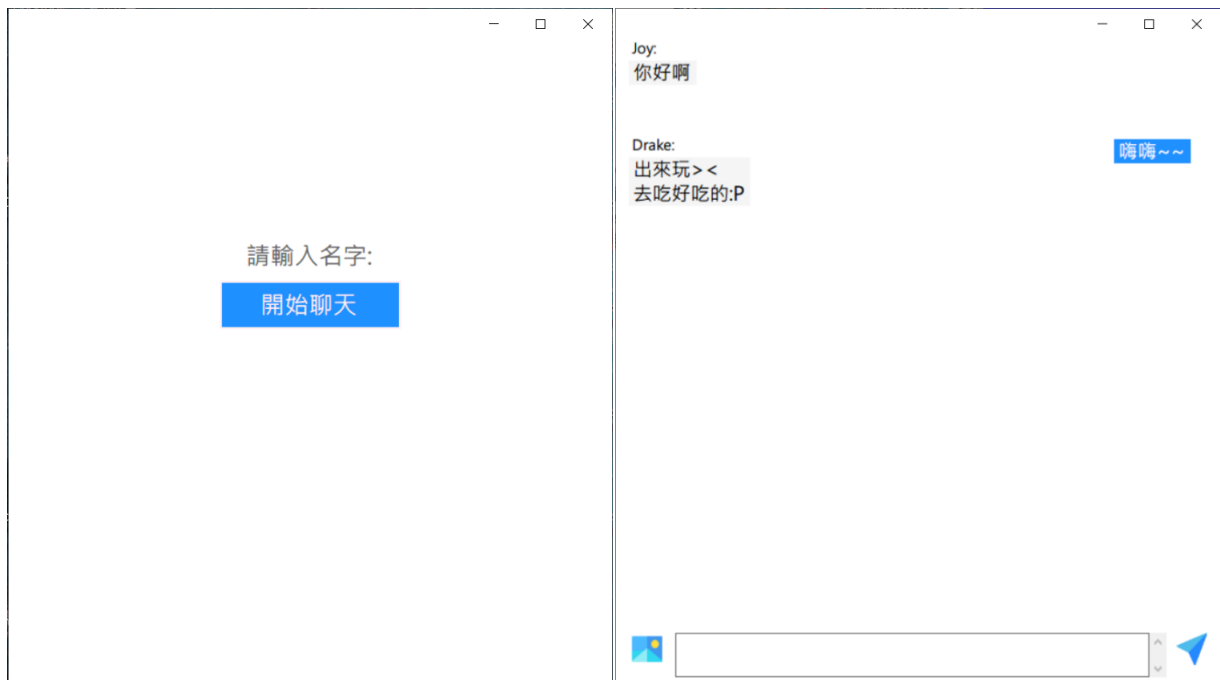
        if (name_[i] == myname)
        {
            l1[i].Visible = false;
            l2[i].BackColor = Color.DodgerBlue;
            l2[i].ForeColor = Color.White;
            int Width = splitContainer1.Panel1.Width - l2[i].Width - 40;
            p[i].Location = new Point(Width, (Int32)height);
        }

        Console.WriteLine("i = " + i + ",p[i].Location = " + p[i].Location);
    }
}
```

10. 關閉表單時，通知伺服器端、關閉 socket 與監聽執行緒

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    Send("9");
    s.Close();
    if(Th != null) Th.Abort();
}
```

11. 用戶端初始畫面與聊天畫面



C. 防錯

1. 伺服器離線，用戶端會跳出通知並建議用戶嘗試重新連線
2. 伺服器斷線，用戶端會跳出通知並回到初始畫面
3. 用戶端若沒有輸入名字，按下開始也不會連線
4. 用戶端若沒有輸入訊息，按下傳送也不會有反應