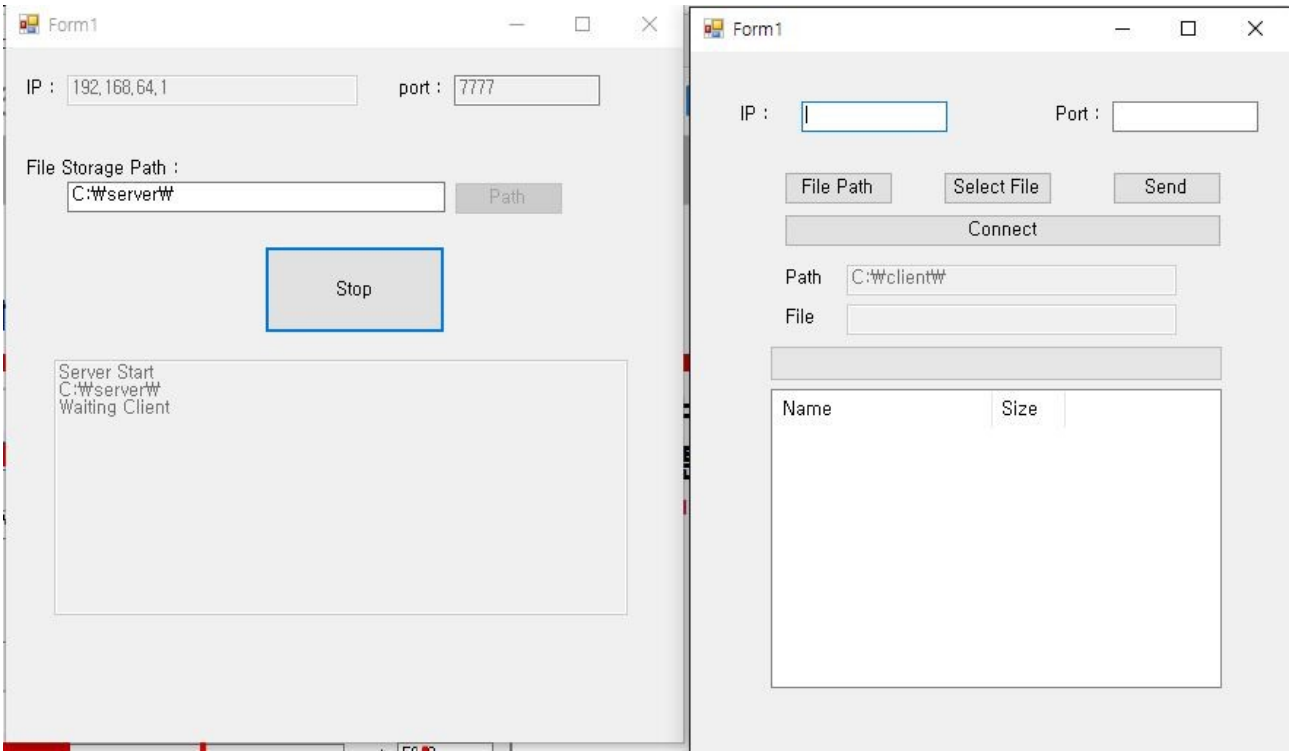
소프트웨어실습 3

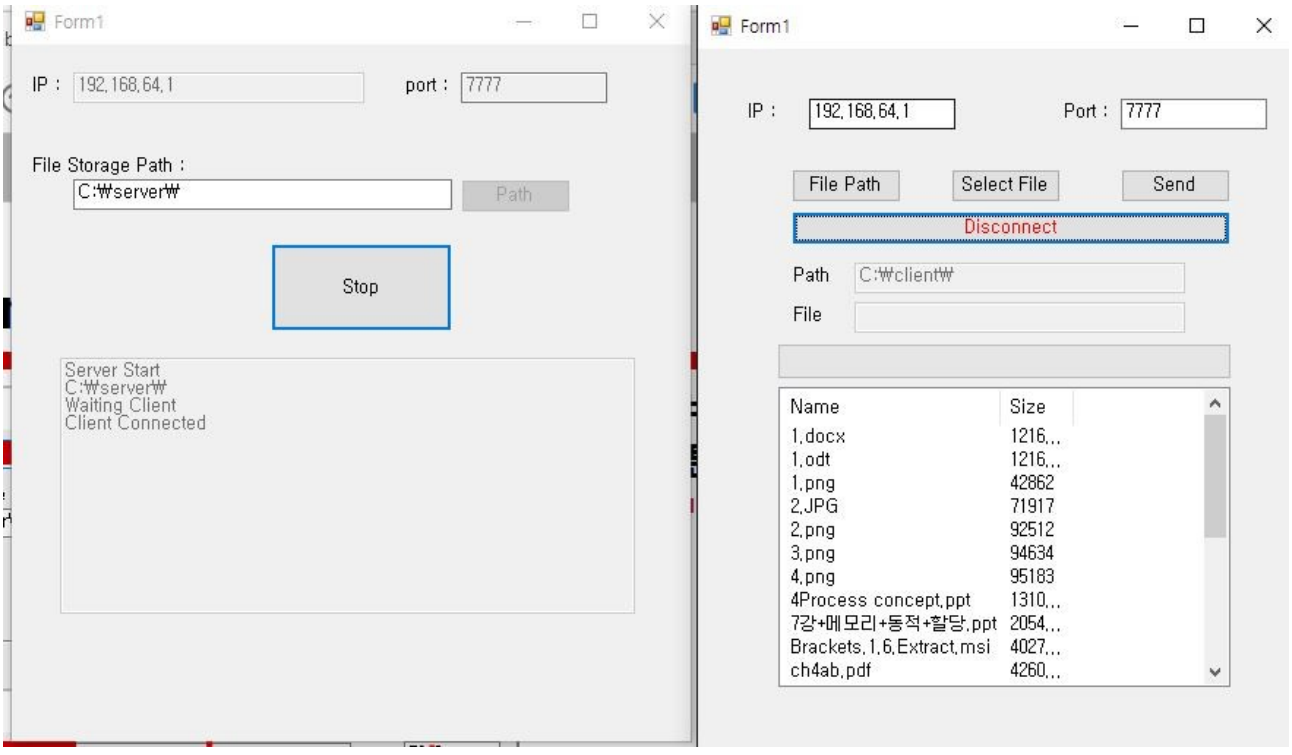
과제 2

2013726003
컴퓨터소프트웨어학과
임우재

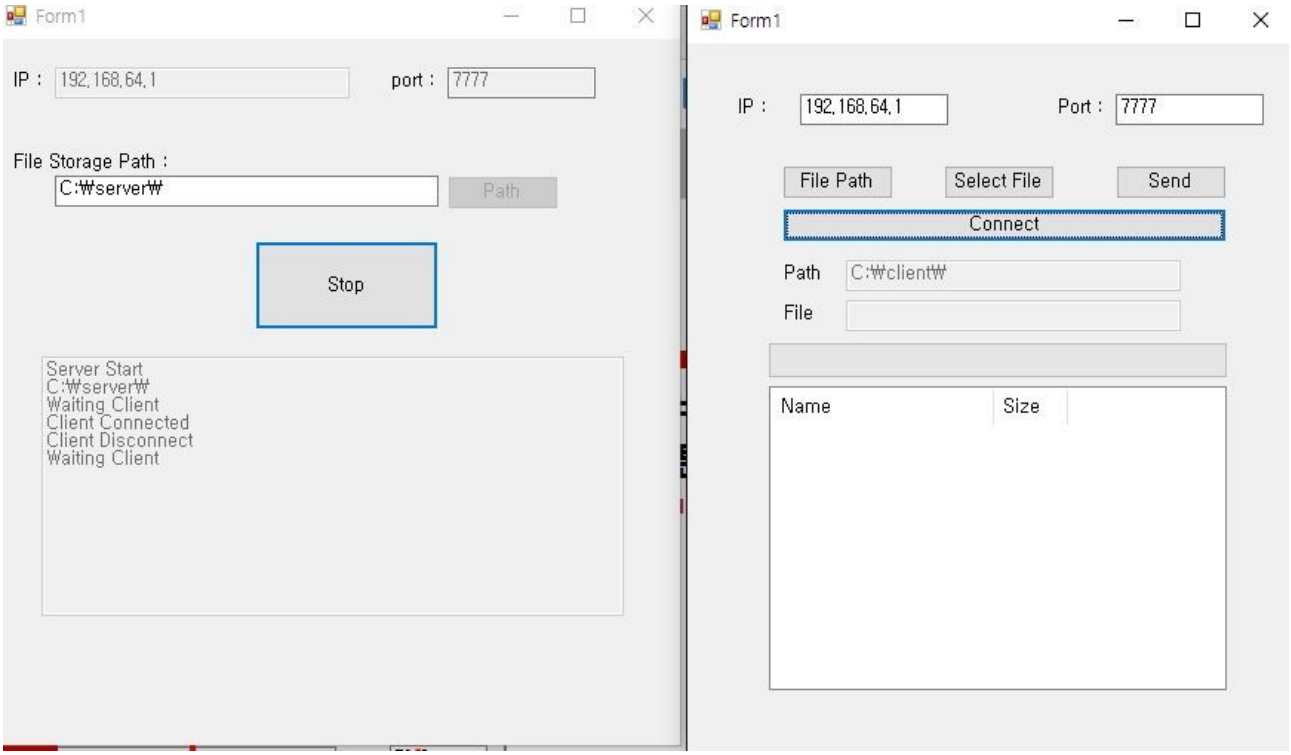
1. 서버가 클라이언트를 기다리는 중



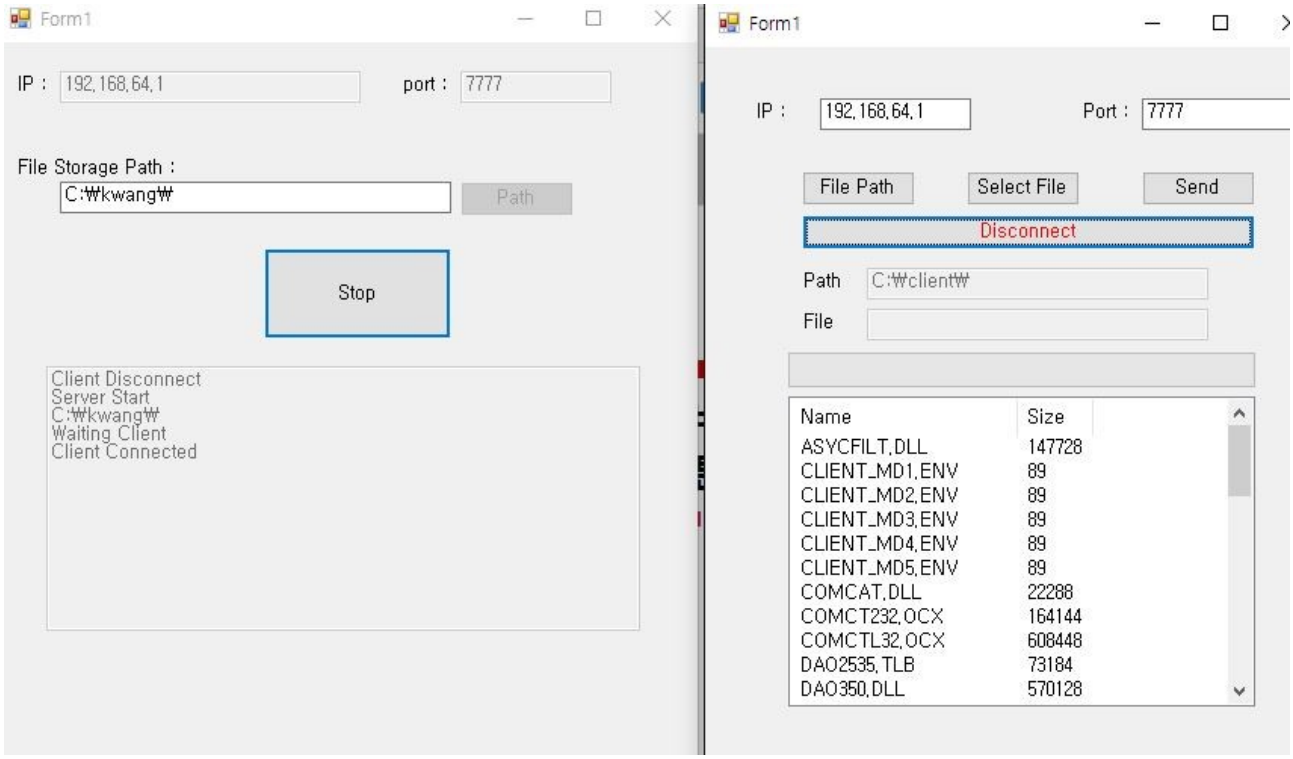
2. 클라이언트 접속



3. 클라이언트가 연결을 끊음



4. 서버를 끈다음 서버의 패스를 바꾼다음 재연결



1. 1~4 까지의 코드 설명

1.1 서버의 코드

```
if(btn_start.Text == "Start")
{

    Btn_Path.Enabled = false;
    btn_start.Text = "Stop";
    append_log("Server Start");
    append_log(FSPTb.Text);
    this.m_Thread.Start();
    Btn_Path.Enabled = false;
    PortTb.Enabled = false;
}
```

스타트 버튼이 눌리면 Path 버튼과 Port 라벨이 유저와 상호작용을 못하게 했다.

```
public void RUN()
{
    int c = 0;
    this.m_Listener = new TcpListener(IPAddress.Any, Int32.Parse(PortTb.Text));
    this.m_Listener.Start();
    System.IO.DirectoryInfo di = new System.IO.DirectoryInfo(FSPTb.Text);
```

서버가 시작되면 서버가 지정한 경로의 정보를 받아온다.

```
if(Client.Connected)
{

    this.m_blsClientOn = true;
    append_log("Client Connected₩n");
    this.m_NetStream = new NetworkStream(Client);

    foreach (System.IO.FileInfo f in di.GetFiles())
    {

        Packet packet = new Packet();
        packet.filename = f.Name;
        packet.filesize = f.Length;
        Packet.Serialize(packet).CopyTo(this.sendBuffer, 0);
        this.m_NetStream.Write(this.sendBuffer, 0, this.sendBuffer.Length);
        this.m_NetStream.Flush();
        sendBuffer.Initialize();
    }
}
```

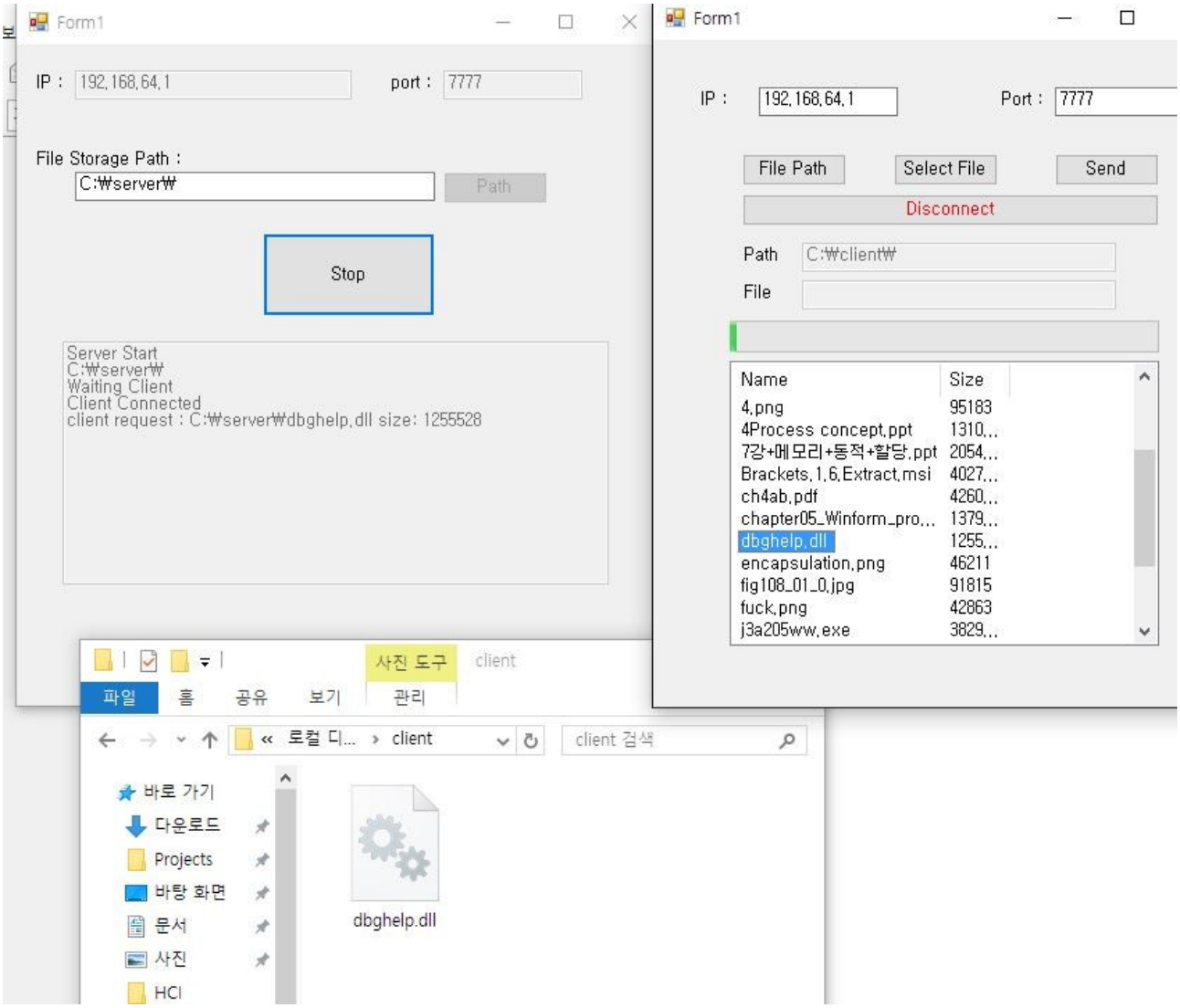
그후 클라이언트가 연결되면 foreach 문을 사용하여 디렉토리에 있는 파일의 갯수만큼 m_NetStream 에 write 한다.

1.2 클라이언트의 코드

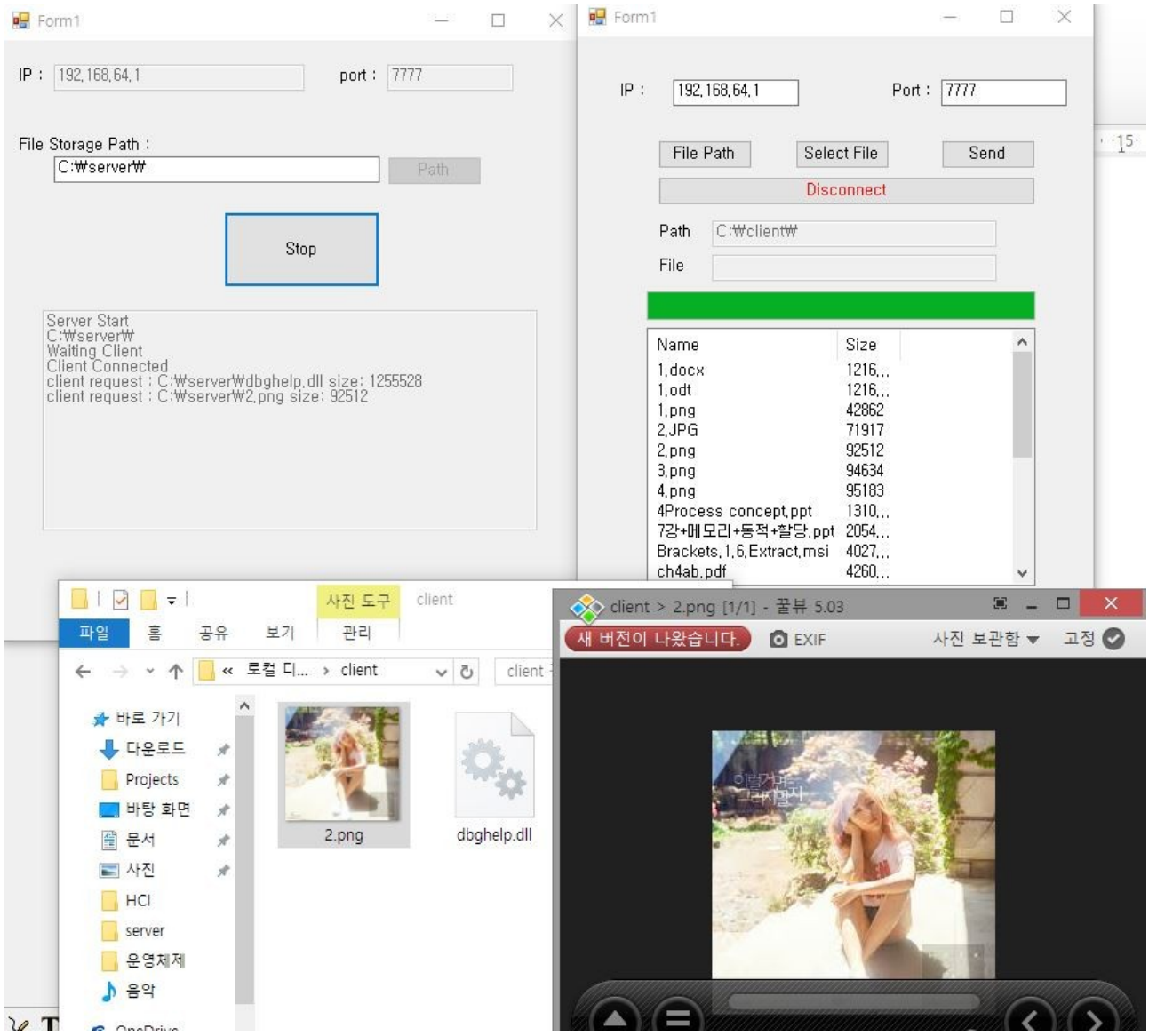
```
public void run()
{

    this.connection = true;
    this.m_NetStream = this.client.GetStream();
    int c = 0;
    while (connection == true)
    {
        this.m_NetStream.Read(this.readBuffer, 0, 1024 * 5);
        Packet packet = (Packet)Packet.Deserialize(readBuffer);
        readBuffer.Initialize();
        if (packet.type == 0)
        {
            listView1.Items.Add(new ListViewItem(
               new string[]{
                packet.filename, (packet.filesize).ToString()
            }
            ));
        }
```

클라이언트는 읽어들인 패킷이 1 이면 리스트뷰 아이템에 해당 패킷 정보를 추가한다.

5. 클라이언트가 서버에 있는 파일들중 하나를 다운로드 함



6. 다운로드가 잘됨


Form1
IP : 192.168.64.1
port : 7777
File Storage Path :
C:₩server₩
Path
Stop
Server Start
C:₩server₩
Waiting Client
Client Connected
client request : C:₩server₩dbghelp.dll size: 1255528
client request : C:₩server₩2.png size: 92512
Form1
IP : 192.168.64.1
Port : 7777
File Path
Select File
Send
Disconnect
Path C:₩client₩
File
Name Size
1.docx 1216...
1.odt 1216...
1.png 42862
2.JPG 71917
2.png 92512
3.png 94634
4.png 95183
4Process concept.ppt 1310...
7강+메모리+동적+할당.ppt 2054...
Brackets.1.6.Extract.msi 4027...
ch4ab.pdf 4260...
사진 도구
client
파일
홈
공유
보기
관리
« 로컬 디... › client
바로 가기
다운로드
Projects
바탕 화면
문서
사진
HCI
server
운영체제
음악
2.png
dbghelp.dll
client > 2.png [1/1] - 꿀뷰 5.03
새 버전이 나왔습니다.
EXIF
사진 보관함
고정

2. 5~6 코드설명

2.1 클라이언트 코드

```
private void listview_dc(object sender, EventArgs e)
{
    Packet packet2 = new Packet();
    packet2.type = 1;
    packet2.filename = listView1.FocusedItem.Text;
    Packet.Serialize(packet2).CopyTo(this.sendBuffer, 0);
    this.m_NetStream.Write(this.sendBuffer, 0, this.sendBuffer.Length);
    this.m_NetStream.Flush();
    for (int i = 0; i < 1024 * 5; i++)
        this.sendBuffer[i] = 0;
}
```

리스트뷰가 더블클릭되면 해당 아이템의 이름을 가진 패킷을 서버에게로 전달

이때 보내는 패킷의 타입은 1

2.2 서버 코드

```
while(this.m_blsClientOn)
{
    try
    {
        nread = 0;
        nread = this.m_NetStream.Read(this.readBuffer, 0, 1024 * 5);
        Packet packet2 = new Packet();
        packet2 = (Packet)Packet.Deserialize(readBuffer);
```

읽어들인 패킷의 타입이 1 이라면

```
else if (packet2.type == 1)
{
    Packet packet3 = new Packet();
    packet3.filename = packet2.filename;
    packet3.type = 1;
    filedata = File.ReadAllBytes(FSPTb.Text + packet3.filename);
    packet3.filesize = filedata.Length;
    append_log("client request : " + FSPTb.Text + packet3.filename + " size: " + packet3.filesize.ToString());

    Packet.Serialize(packet3).CopyTo(this.sendBuffer, 0);
    this.m_NetStream.Write(this.sendBuffer, 0, this.sendBuffer.Length);
    this.m_NetStream.Flush();

    sendBuffer.Initialize();

    int count = (int)((packet3.filesize / (1024 * 4)) + 1);
```

filedata 에 요청한 파일의 바이트를 모두 읽어온 다음에
클라이언트에 요청한 파일의 이름과 사이즈를 다시 보내주고
파일사이즈에 따라 몇 번 나눠서 보낼껀지 count 값에 저장한다.

```
if (i != (count - 1))
{
    for (int k = i * (1024 * 4); k < (i * (1024 * 4)) + (1024 * 4); k++)
    {
        packet3.filedata[k - (i * 1024 * 4)] = filedata[k];
    }
}

else if(i == (count-1))
{
    packet3.filedata.Initialize();
    for (int k = i * (1024 * 4); k < ((i * (1024 * 4)) + ((int)packet3.filesize % (1024 * 4))); k++)
        packet3.filedata[k - (i * 1024 * 4)] = filedata[k];
    filedata.Initialize();
}

packet3.type = 2;
this.send_Thread = new Thread(delegate()
    {
        SendPacket(packet3);
    });
send_Thread.Start();
```

그후 보낼패킷에 파일데이터를 잘라서 넣어준다음에 센드 스레드를 실행시켜준다.
이때 보내는 패킷의 타입은 2 이다.

```
bool a;
while(true)
{
    a = send_Thread.IsAlive;
    if (a == true)
        Thread.Sleep(1);
    else
        break;
}
```

그후 센드 스레드와 항상 읽고 있는 m_thread 와의 동기화 처리를 해주었다.

2.3 클라이언트 코드

```
else if(packet.type == 1)
{
    c = 0;
    count = ((int)packet.filesize / (1024 * 4)) + 1;
    size = (int)packet.filesize;
    FileStream wr = File.OpenWrite(textBox_Path.Text + packet.filename);
    progressBar1.Value = 0;
    progressBar1.Step = 1;
    progressBar1.Minimum = 0;
    progressBar1.Maximum = count;
    wr.SetLength(packet.filesize);
    wr.Close();
    progressBar1.Focus();

}
```

이후 1 번타입의 패킷이 도착하면 클라이언트는 파일을 받을 준비를 한다.

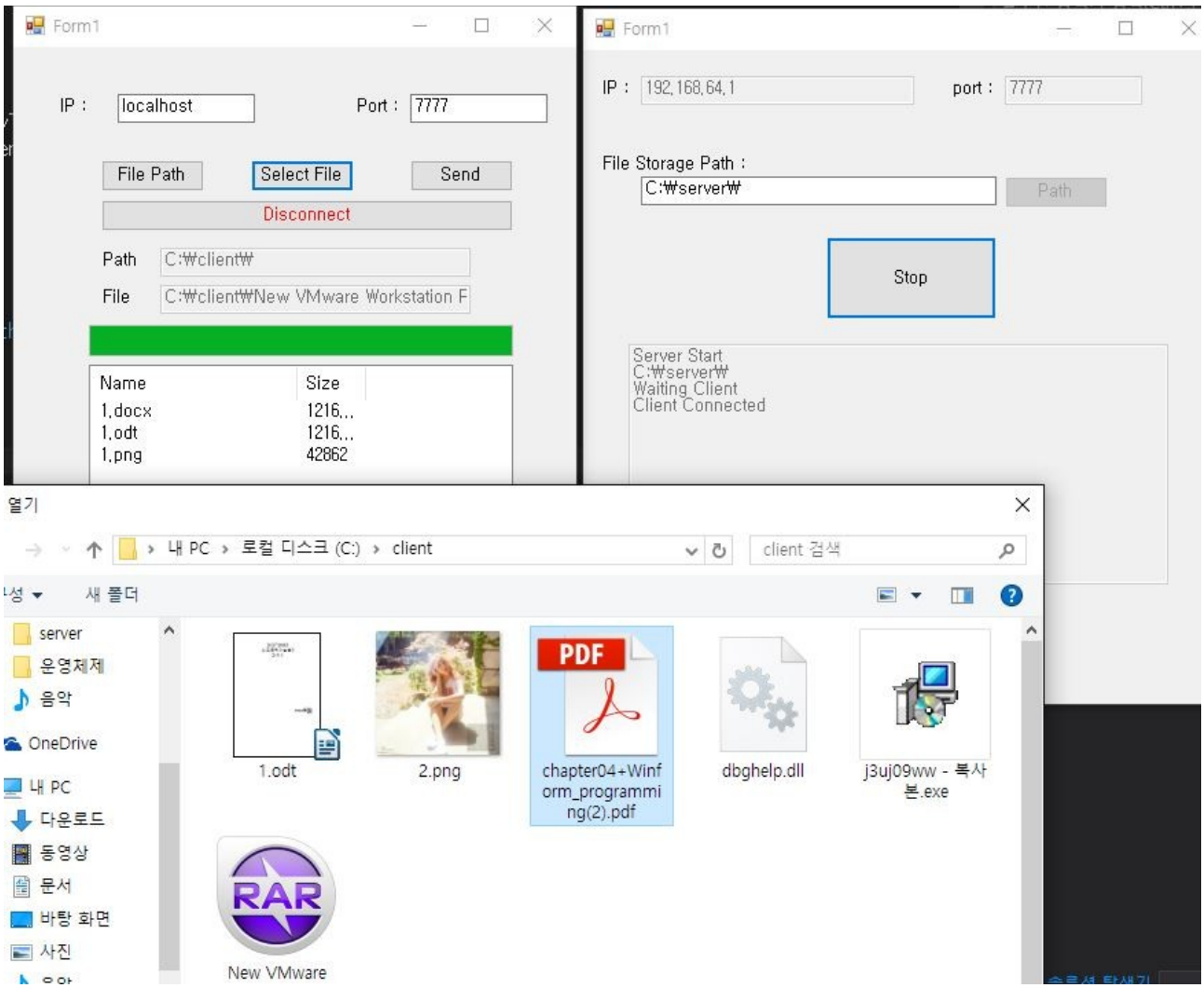
파일을 생성하고 파일의 길이를 설정한 다음

파일데이터가 몇번 날라올지 설정한다.

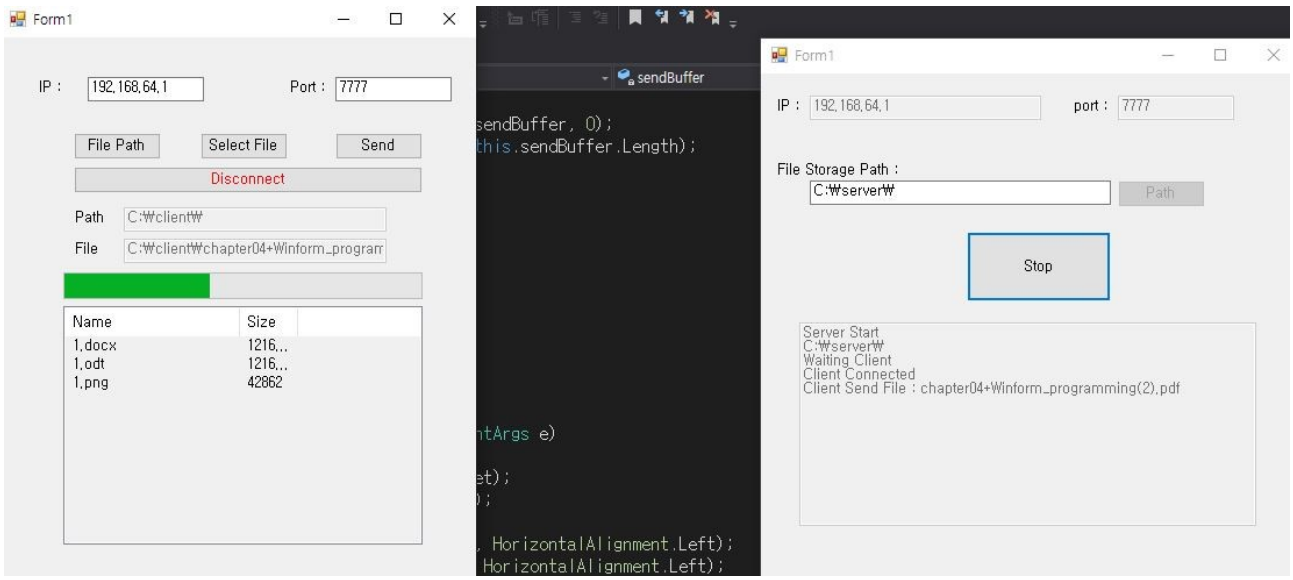
```
else if(packet.type == 2)
{
    FileStream wr = File.OpenWrite(textBox_Path.Text + packet.filename);
    wr.Position = 1024 * 4 * c;
    if (c != (count - 1))
    {
        for (int i = 0; i < packet.filedata.Length; i++)
            wr.WriteByte(packet.filedata[i]);
    }
    if ( c == (count -1))
    {
        for (int i = 0; i < wr.Length % (1024 * 4); i++)
            wr.WriteByte(packet.filedata[i]);
    }
    wr.Flush();
    wr.Close();
    c++;
    packet.filedata.Initialize();
    progressBar1.PerformStep();
    if(c == count)
        listView1.Focus();
}
```

이후 2 번타입의 패킷이 도착하면 파일에 바이트를 쓴다.

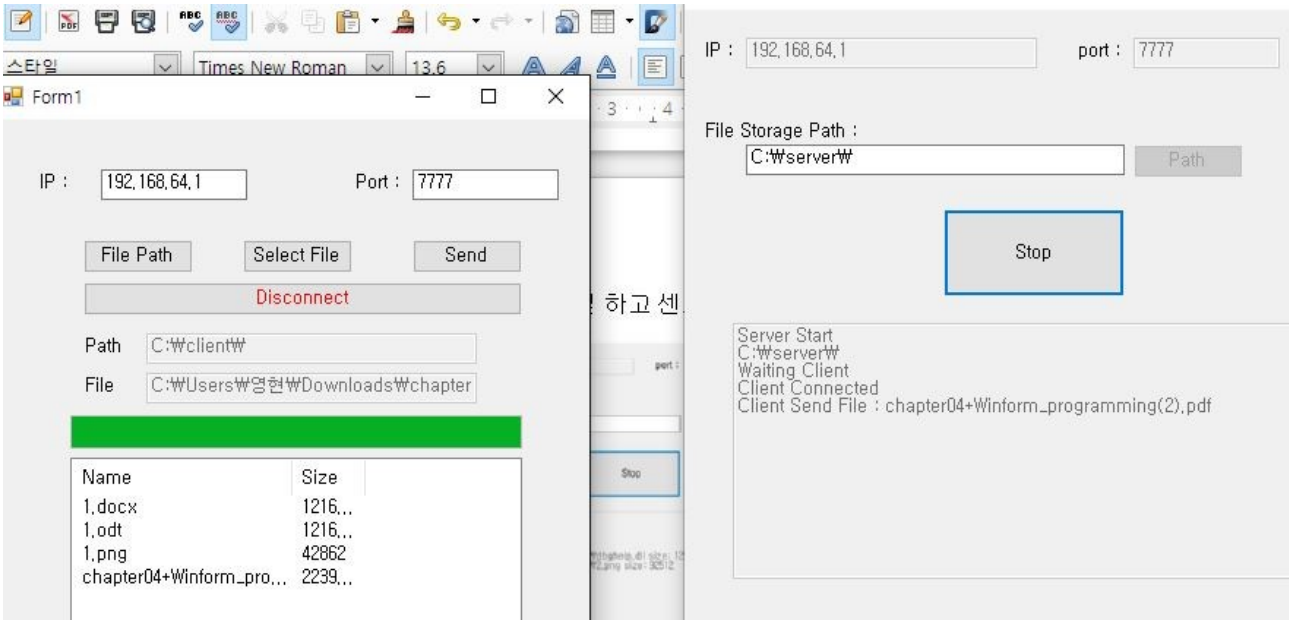
7. 셀렉트 파일 하고 센드 한다.


Form1
IP : localhost
Port : 7777
File Path
Select File
Send
Disconnect
Path C:₩client₩
File C:₩client₩New VMware Workstation F
Name Size
1.docx 1216...
1.odt 1216...
1.png 42862
Form1
IP : 192.168.64.1
port : 7777
File Storage Path :
C:₩server₩
Path
Stop
Server Start
C:₩server₩
Waiting Client
Client Connected
열기
내 PC › 로컬 디스크 (C:) › client
client 검색
새 폴더
server
운영체제
음악
OneDrive
내 PC
다운로드
동영상
문서
바탕 화면
사진
1.odt
2.png
chapter04+Winform_programming(2).pdf
dbghelp.dll
j3uj09ww - 복사본.exe
New VMware

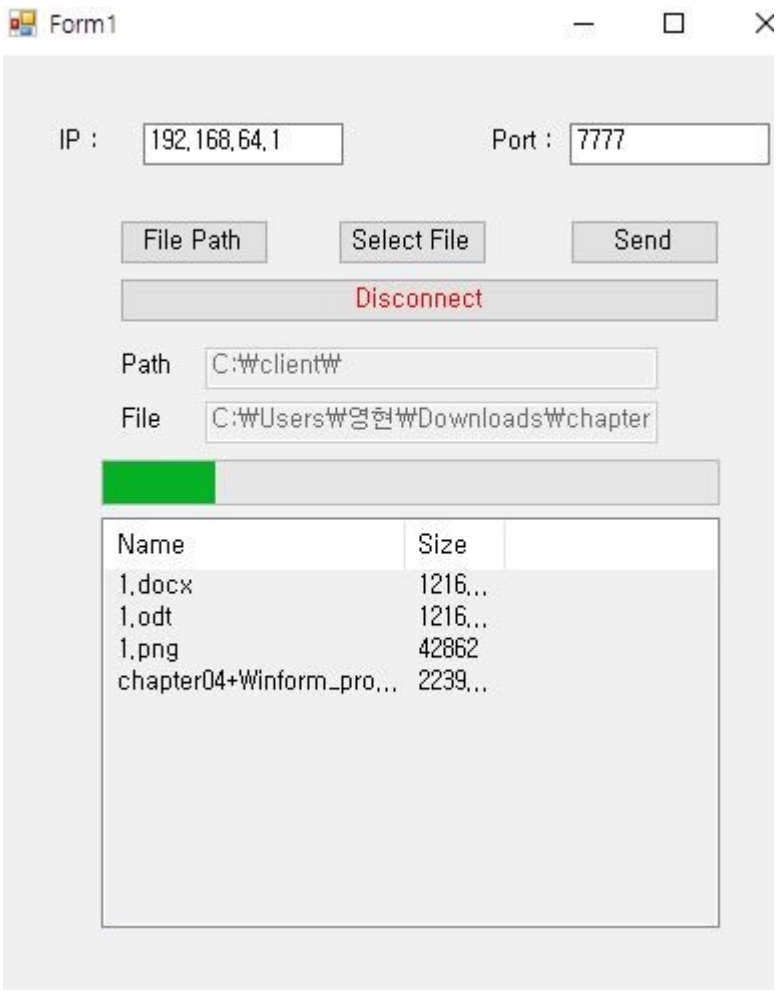
센드버튼 클릭


Form1
IP : 192.168.64.1
Port : 7777
File Path
Select File
Send
Disconnect
Path C:₩client₩
File C:₩client₩chapter04+Winform_program
Name Size
1.docx 1216...
1.odt 1216...
1.png 42862
sendBuffer
sendBuffer, 0);
this.sendBuffer.Length);
ntArgs e)
et);
, HorizontalAlignment.Left);
HorizontalAlignment.Left);
Form1
IP : 192.168.64.1
port : 7777
File Storage Path :
C:₩server₩
Path
Stop
Server Start
C:₩server₩
Waiting Client
Client Connected
Client Send File : chapter04+Winform_programming(2).pdf

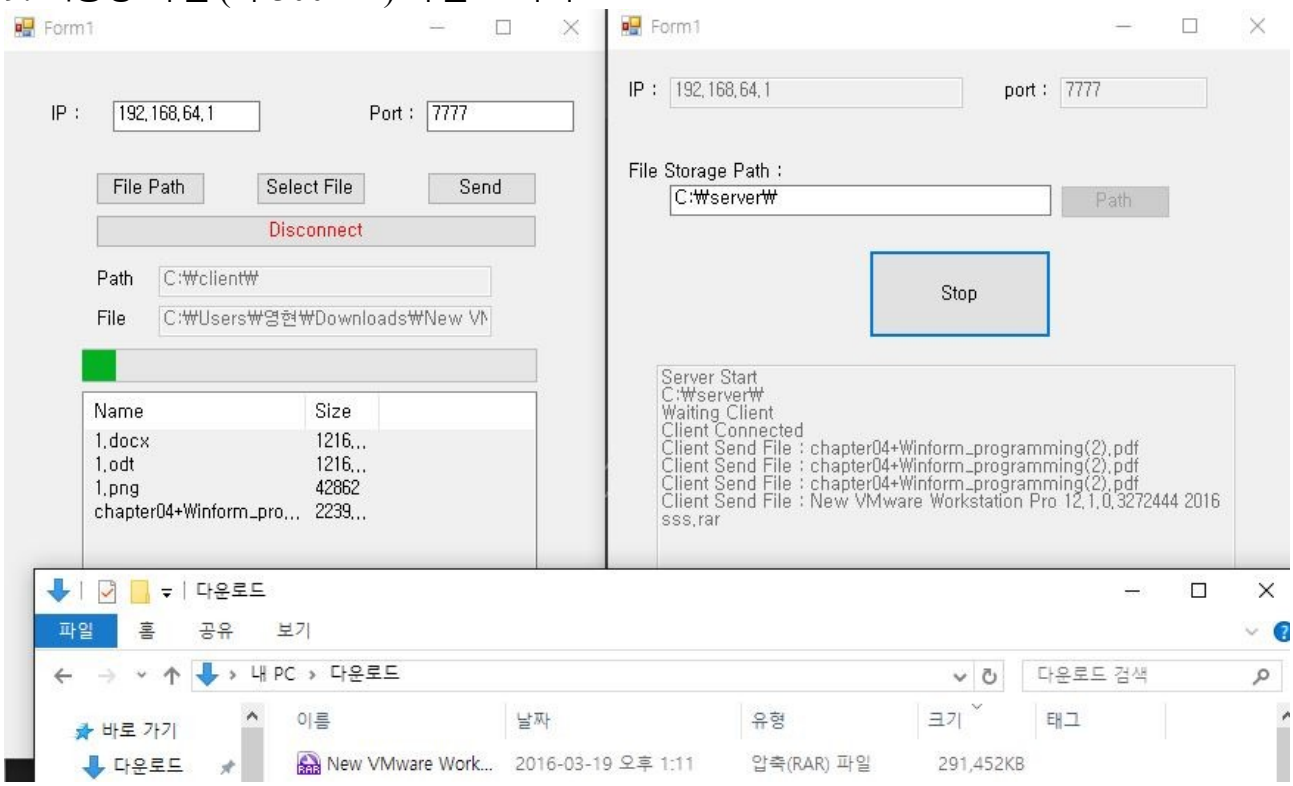
센드가 완료된뒤 목록이 갱신되는 모습



8. 센드 도중에는 리스트 뷰 상호작용이 불가능하다.



9. 대용량 파일 (약 300MB) 파일 보내기



3. 7~9 코드 설명

```
private void btn_Send_Click(object sender, EventArgs e)
{
    try
    {
        FileInfo filedata = new FileInfo(label_File.Text);
        int count = ((int)filedata.Length / (1024 * 4)) + 1;
        progressBar1.Value = 0;
        progressBar1.Step = 1;
        progressBar1.Minimum = 0;
        progressBar1.Maximum = count;
        progressBar1.Focus();
        listView1.Enabled = false;

        send_Thread.Start();

    }
    catch
    {
        MessageBox.Show("File is not Selected");
    }
```

send 버튼이 클릭되면 프로그레스 바가 설정된뒤 send_Thread 가 실행된다.

send_Thread 에서는

```
public void SendPacket()
{
    Thread.BeginCriticalRegion();
    byte[] filedata = File.ReadAllBytes(textBox_File.Text);
    int count = (filedata.Length / (1024 * 4)) + 1;
    Packet packet3 = new Packet();
    FileInfo fi = new FileInfo(textBox_File.Text);
    packet3.type = 2;
    packet3.filename = fi.Name;
    packet3.filesize = fi.Length;
    for (int i = 0; i < count; i++)
    {
        if (i != (count - 1))
        {
            for (int k = i * (1024 * 4); k < (i * (1024 * 4)) + (1024 * 4); k++)
            {
                packet3.filedata[k - (i * 1024 * 4)] = filedata[k];
            }
        }
```

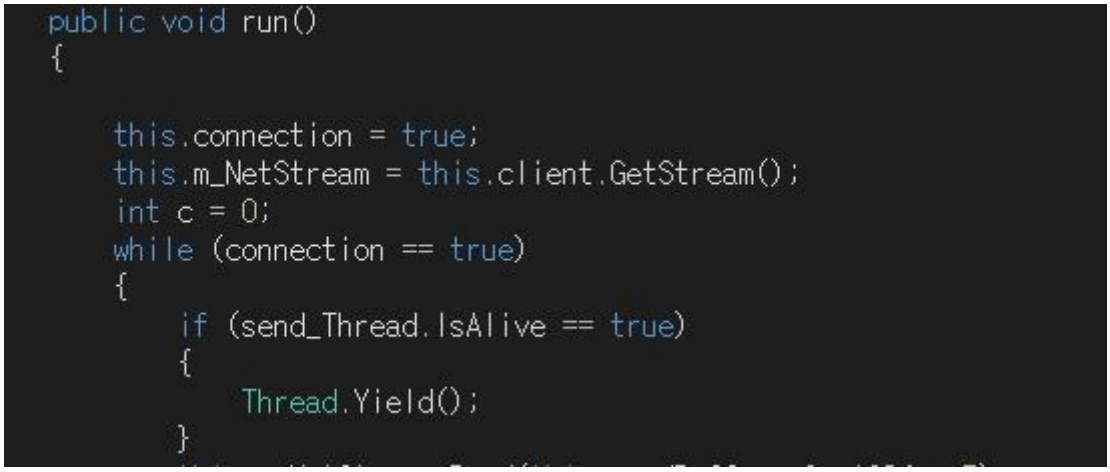
```
            }

            if (i == (count - 1))
            {
                packet3.filedata.Initialize();
                for (int k = i * (1024 * 4); k < ((i * (1024 * 4)) + ((int)packet3.filesize % (1024 * 4))); k++)
                    packet3.filedata[k - (i * 1024 * 4)] = filedata[k];
                filedata.Initialize();
                listView1.Focus();
            }
            packet3.type = 2;
            Packet.Serialize(packet3).CopyTo(this.sendBuffer, 0);
            this.m_NetStream.Write(this.sendBuffer, 0, this.sendBuffer.Length);
            this.m_NetStream.Flush();
            pb_perform();
            sendBuffer.Initialize();
            packet3.filedata.Initialize();
        }
        packet3.type = 0;
        Packet.Serialize(packet3).CopyTo(this.sendBuffer, 0);
        m_NetStream.Write(this.sendBuffer, 0, this.sendBuffer.Length);
        m_NetStream.Flush();
        listView1.Enabled = true;
        listView1.Items.Clear();
        Thread.EndCriticalRegion();
        send_Thread = new Thread(SendPacket);
    }
```

SendPacket()의 함수가 실행되는데 서버에서 패킷을 보내는 방식과 크게 다르지 않다.
몇가지 다른점을 소개하면
Thread.BeginCriticalRegion(); 메소드와
Thread.EndCriticalRegion(); 메소드다.

begin 메소드는 호스트에게 지금 실행하는 것들이 위험한 부분이라고 알려주는 역할을 하고 end 메소드는 이제 끝났다는 것을 알려준다.
또한 pb_perform() 메소드는

```
    public void pb_perform()
    {
        this.Invoke(new MethodInvoker(delegate()
        {
            this.progressBar1.PerformStep();
        }));
    }
```

크로스-스레드 exception 을 막기위해 인보크로 프로그래스바를 진행시킨다.

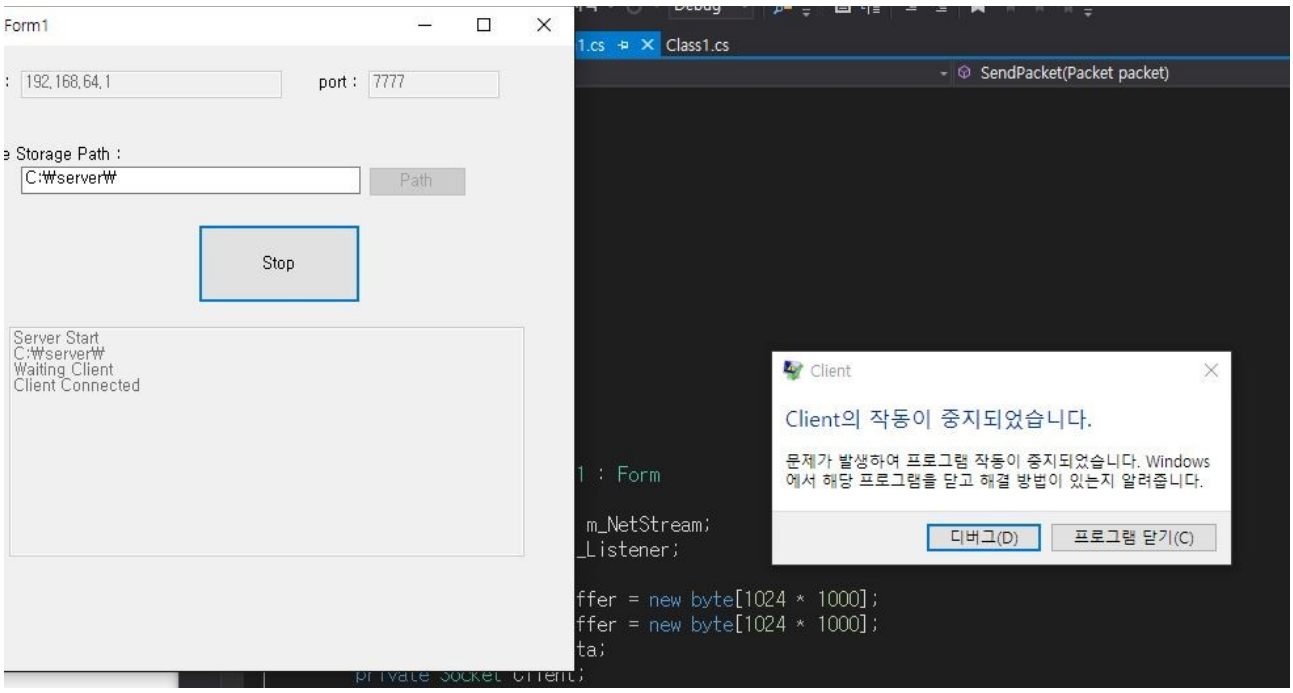


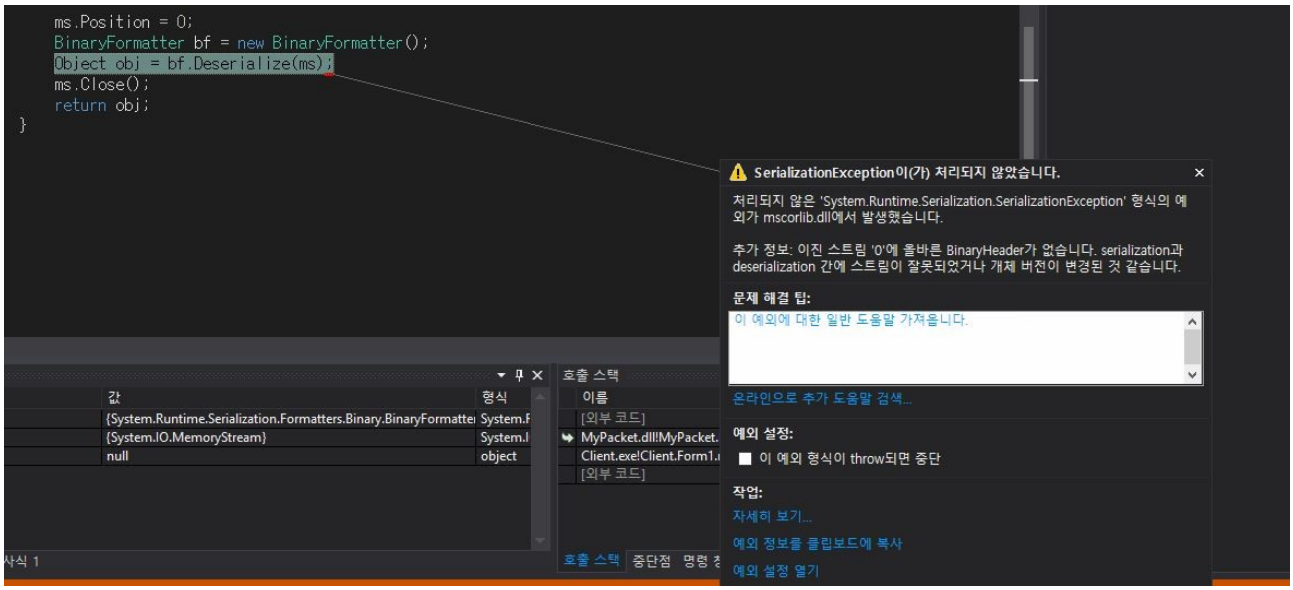
```
public void run()
{

    this.connection = true;
    this.m_NetStream = this.client.GetStream();
    int c = 0;
    while (connection == true)
    {
        if (send_Thread.IsAlive == true)
        {
            Thread.Yield();
        }
```

또한 run 메소드를 실행하는 m_Thread 와의 동기화를 위해
Thread.Yield() 메소드를 추가하였다.
Thread.Yield 는 실행을 기다리는 다른 스레드를 실행시키는 것으로
send_Thread 가 현재 실행되고 있으면 다른 스레드를 실행시키도록 하였다.

4. 고찰

이번 과제는 서버는 서버를 시작할 경우 RUN 메소드를 m_Thread 가 실행시키도록 하고 클라이언트는 run 메소드를 m_Thread 가 실행시키도록 했다. 그후 서버는 항상 RUN 메소드 에서네트워크스트림을 읽고 들어온 타입에 대하여 일을 수행하면 됬기에 크게 구현이 어렵지 않았지만 클라이언트 같은 경우는 run 메소드에서 항상 읽고 있기도 하면서 SEND 버튼을 클릭할 경우 바로 Send_Thread 가 실행되어야 해서 조금 골치 아팠다. 그 외에는 파일데이터를 한번에 보낼수 없어서 짤라서 보내는 부분에서 많이 헤맸던 것 같다.
현재 대용량의 파일을 보낼때 속도가 많이 느려서 이부분을 해결해보려고 했는데 이런 에러가 나면서 안됬다.





이 문제를 해결해볼려고 구글에서도 보고 msdn 에도 들어가봤지만 msdn 에는 serialize 나 deserialize 할때 스로우되는 익셉션이라 라는 외의 설명은 없고 구글에서도 해결법을 찾지 못하였다. 오직 readbuffer 와 sendbuffer 의 크기를 키웠을 때만 발생한다...

또한 연결되어있는 도중에 한쪽이 끊기게 될 경우도 대비를 해두었다. 파일 전송중에 끊기더라도 try-catch 로 프로그램이 바로 꺼지지 않게 해두었다.