

```

    bool _pending = false;
    SocketAsyncEventArgs _sendArgs = new SocketAsyncEventArgs();
    SocketAsyncEventArgs _recvArgs = new SocketAsyncEventArgs();

    public void Start(Socket socket)
    {
        _socket = socket;

        _recvArgs.Completed += new EventHandler<SocketAsyncEventArgs>(OnRecvCompleted);
        _recvArgs.SetBuffer(new byte[1024], 0, 1024);

        _sendArgs.Completed += new EventHandler<SocketAsyncEventArgs>(OnSendCompleted);

        RegisterRecv(_recvArgs);
    }

    public void Send(byte[] sendBuff)
    {
        lock (_lock)
        {
            _sendQueue.Enqueue(sendBuff);
            if (_pending == false)
                RegisterSend();
        }
    }

    public void Disconnect()
    {

```

지정된 데리어스트는 `_recvArgs`와 `_sendArgs`

의 리턴값

Callback의 queue를 보면 사용.

→ 디蹲에 대해서는 학습하고 있으나
체결화의 예제

`_sendArgs`, `Bufferlist` = `Byte`를 `List<byte[]>`로 받!

or) 훌륭한 사용법 예제
`_sendArgs`, `SetBuffer`

White (`_SendQueue.Count > 0`)

} `byte[] buff = _SendQueue.Dequeue()`

`_SendArgs`, `Bufferlist`.`Add`(new `ArraySegment<byte>(buff, 0, length))`

```

    SocketAsyncEventArgs _sendArgs = new SocketAsyncEventArgs();
    _socket.Close();

    #region 네트워크 통신

    void RegisterSend()
    {
        _pending = true;

        List<ArraySegment<byte>> list = new List<ArraySegment<byte>>();

        while (_SendQueue.Count > 0)
        {
            byte[] buff = _SendQueue.Dequeue();
            list.Add(new ArraySegment<byte>(buff, 0, buff.Length));
        }

        _SendArgs.BufferList = list;

        bool pending = _socket.SendAsync(_SendArgs);
        if (pending == false)
            OnSendCompleted(null, _SendArgs);
    }

    void OnSendCompleted(object sender, SocketAsyncEventArgs args)
    {
        lock (_lock)
        {
            if (args.BytesTransferred > 0 && args.SocketError == SocketError.Success)

```

Arrayof Segment

= Array의 링크.

→ Struct / Stack 구조

C++의 경우 배열 사용

GoGPt / Object Oriented

C++은 포인터를 이용

C++은 pointer를 통해 index를 참조함.

조심 : Bufflist 를 만들고 사용하기 = 3 네트워크 할.

위의 List (ArraySegment) 2 개 이상에 있고 pendingList 를 두자.
↓

Pending 을 통해 Indirect 정의 가능.

- SendArgs . BuffList = null = PDI

- pendingList . Clear()

→ Console . Writeline ("Transferred bytes: " + _sendargs.BytesTransferred)
Output 뭔가?

PendingList 를 통한 bool 가능.

Send → Lock하고 → 데이터 넣고 → ~~330000~~ Queue.
↳ queue registerSend

registerSend → Dequeue → PendingList . Add (ArraySegment)

→ BuffList = PendingList → OnSendCompleted()

OnSendComp → bytesClear → if (Count > 0) { registersent }

모든 패킷 헤더가 모였나? → Sendqueue의 bytes의 체크하기.

1. 일정한 시간에 끊기 않는 패킷이라면 재전송 필요!

제일로 궁금해하는 헬렌과... → 대구자체를 뛰어넘어야 하는 경주 성장.

도와보내는 대구 → 시비현장에서 하늘에 만드는거? = 농식 사육.