|  |  |
| --- | --- |
| 2018/10/23 | 일부 작성시키고 테스트한 라이브러리 코드를 처음 커밋했다.  서버-클라 사이에 테스트용 패킷은 MessagePacket을 주고 받는 것을 테스트 하였다. 잘 동작한다. 패킷 처리시 아직 동시성 문제는 해결하지 않았다. 잘 알려진 Job처리를 넣을 예정. 추후 구글프로토버퍼를 붙일 예정이다.  ServerNetWorkDepartment 가 서버를 작동하는 클래스이다.  - \*\*\*Processor 라고 이름 붙여진 멤버들은 GQCS의 Network-IO 이벤트를 처리 하기 위해 존재한다.  - \*\*\*Processor 중 Send, Recv는 Session 클래스에서 Pointer로 가지고 있다. Send, Recv 는 Session이 처리하도록 고민해봄직 하다. 너무 타고들어가는 느낌이 있다.  ThreadManager  - 생성된 Thread들을 Department 라는 enum을 두어 저장해두고 있다.  - ThreadId는 TLS 로 저장해두고 있다.  - Server 클래스와 , PacketProcess 클래스 에서 따로 Thread Array를 멤버로 가지고 있는데 필요가 없을 듯하다. (추후 삭제하자) |
| 2018/10/28 | - SendBufferQueue로 Concurrent\_Queue를 사용해보기 위해 새로 추가하였다.  - 아직 정확하게 동작할 코드는 작성하지 않았다.  SendBufferQueue 로 Send가 정상 동작한다면 그 후에 구글 프로토버퍼를 붙여볼 계획이다. |
| 2018/10/29 | - ThreadDepartment 클래스를 추가 하였다. 해당 클래스는 생성된 Thread들을 unique\_ptr로 관리한다.  ThreadManager 클래스에서 Thread 생성하기전 Department 를 먼저 생성하고, Thread를 생성하여야 동작한다.  이렇게 나눈 이유는 추후 Thread에 Job을 넘길때 어떤 ThreadDepartment 에 넘길지 선택하기 위함이다. |
| 2018/10/30 | - Session 클래스의 sessionId를 발급하기 위한 IdGenerator 싱글톤 클래스를 추가하였다.  Session이 생성될때 sessionId가 발급되고, 해당 ID 는 서버 프로그램이 종료될때까지 바뀌지 않는다.  이전에는 SessionPool에서 Session을 가져올때 ID를 발급해주었는데 이렇게 되면 Lock을 걸어야해서 최초 서버 구동시 아이디를 모두 발급해버리게 수정 하였다.  - Lock 클래스의 LockSafeScope, LockManager 클래스를 삭제하였다. 단순히 recursive\_mutex로 Lock을 걸었던걸 변경하여 shared\_mutex, shared\_lock, unique\_lock 으로 사용하도록 한다. 매크로 추가. |
| 2018/10/31 | - 기존의 ServerNetWorkDepartment 클래스는 이제 NetworkDepartment를 상속받아 구현하게 되었다.  이는 추후 서버 간의 연결 등 각 NetworkDeapartment를 관리하기 위함이다.  ClientNetworkDepartment도 추가해 두었지만 아직 구현 하진 않았다.  이들을 관리하는 클래스는 NetworkDepartmentManager 클래스이며,  Std::vector<NetworkDepartment\*> 로 관리하고 있다. |
| 2018/11/04 | 에코테스트를 진행하였다. Send를 할 때 SendBufferQueue에 우선 저장을 해 둔후 보내도록 작업 해 두었다. 1-recv, 1-send 를 기본으로 하는데.. 큐에 쌓지 않고 바로바로 보내는 방법도 생각해보자. n-send |
| 2018/11/07 | Job 클래스를 추가하였다. 오브젝트별로 쓰레드에 잡을 던져 처리하는 방식을 사용하기 위해 존재하는 Base Class이며, MemFuncJob 클래스가 이를 상속받고 있다. Job queue로는 Concurrent\_queue를 사용해볼 예정이다. |
| 2018/11/11 | 에코 테스트를 하던 도중 memory 오염이 발생해 찾느라 혼났다.  Recv시 안 옮겨도 될 recvBuf\_ 로우 포인터의 위치를 계속해서 옮겨주고  초기화 하는 코드가 없어 이상한곳에 액세스해 미정의 동작을 보았다.  해당코드를 삭제 후 테스트해보니 정상 동작하였다. |
| 2018/11/21 | Flatbuffer를 추가하여 에코 테스트를 진행하였다. 상당히 잘 동작한다. |