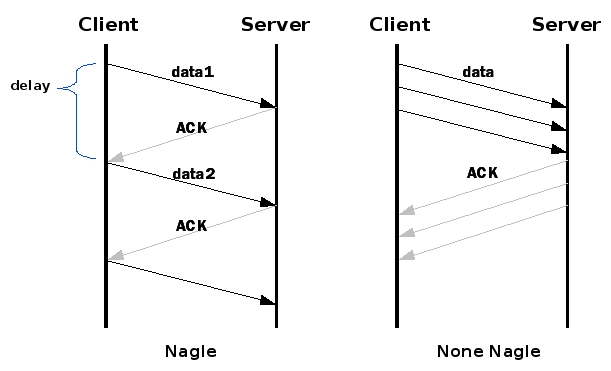
**Ⅰ**

Nagle 알고리즘

Nagle 알고리즘이 적용되면, 운영체제는 패킷을 **ACK**가 오기를 기다렸다가 도착하면, 그 동안 쌓여있던 데이터를 한꺼번에 보내게 된다. 이러한 방식을 사용하게 되면, 대역폭이 낮은 WAN에서 빈번한 전송을 줄이게 됨으로 효과적인 대역폭활용이 가능해진다.

대부분의 경우에 있어서 Nagle 알고리즘은 효율적으로 작동하긴 하지만, 빈번한 응답이 중요한 서비스의 경우에는 적당하지 않은 경우가 발생한다. 예를 들어 X-Terminal을 이용할 경우 마우스 이벤트는 즉시 전달될 필요가 있는데, Nagle알고리즘을 사용하면 아무래도 반응시간이 떨어지게 될 것이다. 실시간적인 반응이 중요한 온라인 게임역시 Nagle 알고리즘을 제거하는게 좋을 것이다.

아래의 이미지는 nagle이 적용되었을 때와 그렇지 않을 때, 어떻게 데이터 전송이 일어나는지를 보여주고 있다.



**[출처]** [[네트워크/MFC] 네이글 알고리즘 및 네이글 알고리즘 적용안하기](http://blog.naver.com/sanglyn/90074781953)|**작성자** [Lynn](http://blog.naver.com/sanglyn)

---------------------------------------------------------------------------------------------------------------------------

■ Nagle 알고리즘이란?

1. 네트워크상의 패킷 수를 줄이기 위해 제안된 알고리즘  
2. ACK를 수신해야만 다음 전송을 진행하는 알고리즘  
 ( 상대방 호스트에서 ACK가 올때 까지 자신의 전송 버퍼 데이터가 있어도 못 전송하는 알고리즘 )

■ Nagle 알고리즘을 적용하지 않았을 때

ACK 수신에 상관없이 자신의 전송 버퍼 데이터가 있으면 바로 상대방 호스트에게 전송하는 알고리즘

■ Nagle 알고리즘 사용방법

예)

level  : IPPROTO\_TCP  
option : TCP\_NODALAY  
   
TRUE  : Nagle OFF (알고리즘을 적용하지 않는다)  
FALSE : Nagle ON  (알고리즘을 적용한다)

server\_sock = socket(PF\_INET SOCK\_STREAM, 0);

option = TRUE; // #define TRUE 1;

// Nagle OFF

setoptsock(server\_sock, IPPROTO\_TCP, TCP\_NODELAY, &option, sizeof(option));

---------------------------------------------------------------------------------------------------------------------------

네트웍 프로그래밍 책 65p

**TCP에서 데이터 전송은 Nagle 알고리즘에 의하여 처리된다.** Nagel 알고리즘에 의하여 전송할 데이터의 누적된 크기가 일정 크기 이상이 되거나 전송을 시도한 이후에 일정 시간이 경과된 뒤에야 데이터가 전송된다. 또한 수신측에서도 데이터 도착을 확인하는 패킷(ACK)을 전송하는 것을 특정 시간 동안 지연시켜서 이 패킷과 같이 전송할 데이터를 기다리게 된다. 패킷을 작은 데이터로 여러 번 전송하는 것은 비효율적일 뿐만 아니라 에러 체크나 전송 확인(ACK)때문에 부하가 증가하는 원인이 된다.

이러한 Nagle algorithm 때문에 TCP 프로토콜은 메시지 중심의 프로토콜처럼 보낸 크기대로 데이터를 수신할 수 없는 것이다.

---------------------------------------------------------------------------------------------------------------------------

나 나름의 정리를 해보자면.. 꾸잉꾸잉

원래 TCP 전송 알고리즘은 송신측이 데이터를 보내고 수신측이 데이터가 수신되었음을 알려오면(ACK 패킷) 다음 패킷을 전송하는 방식.

그런데 이렇게 패킷 한 개 보내고 그 패킷의 수신을 체크하고 그 다음 패킷 한 개 보내고 그 패킷의 수신을 체크하는 방법은 비효율적인거야.

그래서 슬라이딩 윈도우 기법이 나온거. 일정 크기의 윈도우가 있는데 송신측이 데이터를 전송하면 윈도우 크기가 작아지고 ACK 패킷을 수신하면 그만큼 윈도우 크기가 늘어나는 식. 윈도우 크기 범위안에서는 ACK 패킷의 수신에 관계없이 데이터를 연속으로 여러번 전송이 가능하고 윈도우 크기만큼 데이터를 보냇는데(즉 윈도우 크기가 0이 되었는데) ACK 패킷을 수신하지 못하면 더이상 데이터는 전송 못한다는...

그런데 이렇게 되니까 송신측에서 어떤 크기의 데이터를 보내든간에 수신측에서 데이터를 작은 크기로 받게되면 윈도우 크기가 작아지는 행위가 너무 자주 발생해서 데이터 전송의 효율이 떨어지는거얌.(이 부분은 고찰이 필요함????????????) 이걸 실리 윈도우 증후군이라고 부르는거고.

그래서 실리 윈도우 증후군을 피하려고 전송할 데이터의 누적된 크기가 일정 크기 이상이 되거나 전송을 시도한 후에 일정 시간이 경과된 뒤에야 데이터가 전송되게 한거임. 그리고 수신측에서도 데이터 도착을 확인하는 ACK 패킷을 전송하는 것을 특정 시간 동안 지연시켜서 이 패킷과 같이 전송할 데이터를 기다리게 된다는....

**Ⅱ**

네... I'M에 들어와서 쓰는 첫번째 블로그 포스팅입니다...(너무 늦은감이 있지만.. 자료준비때문에 시간조절을 잘 못했네요,,,, 이런게 처음이라서요,,)

여러모로 부족한 내용이지만 너그러운 마음으로 용서해주세요~

﻿

제가 이번에 들고온 주제는 온라인게임!!!!!(우어)중에서도 mmorpg의 역사와 정의 및 특징을 주제로 잡았습니다. 왠만한 남자 or 여자 분들께서 온라인 게임을 한번쯤은 해보셨으실텐데요...  ~~정작 게임의 역사, 장르 등을을 모르시고 게임을 하는 경우가 많아서~~(사실 제가 게임을 좋아합니닷....ㅋㅋㅋ)이번 포스팅을 준비하게 됬습니다.,,,,



월드 오브 워크래프트(wow) 사진으로 시작해보겠습니다.

|  |  |
| --- | --- |
| |  | | --- | | **MMORPG의 역사**  http://ncc.phinf.naver.net/ncc01/2011/6/15/200/7px.jpg  MMORPG를 알기 위해서는 먼저 그 근간인 RPG(Role Playing Game)에 대해 알아야 합니다. 컴퓨터가 도입되기 전의 RPG는 여러 명의 사용자가 테이블에 둘러 앉아 펜, 종이, 주사위를 가지고 즐기던 보드게임이었는데요, GM(게임 마스터)이 게임의 규칙을 결정하고 하나의 세계를 만들어 놓으면, 사용자들은 진행자가 만들어놓은 세계 속에서 각자의 역할을 수행하며 성장해 나가는 것이 기본적인 룰 이었습니다. 일반적인 보드 게임과 달리, 사용자들 간 승패를 가리기 보다는 서로 협동해서 위기를 극복하고 이야기를 순조롭게 이어나가는 데 목적이 있습니다. 테이블 위에서 대사를 주고 받으며 즐긴다고 해서 일본 및 한국에서는 [테이블 롤 플레잉 게임](http://dic.search.naver.com/search.naver?sm=ncc_clk&where=kdic&query=%C5%D7%C0%CC%BA%ED+%B7%D1%C7%C3%B7%B9%C0%D7+%B0%D4%C0%D3&x=34&y=25)(TRPG, Table talk Role playing Game)이라고 부르기도하며, 서양에서는 PnP RPG(Pen-and-Paper RPG)라고 부르기도 합니다. | |



테이블 롤플레잉 게임을 즐기는 모습 <출처: (cc) Diacritica at Wikimedia.org>

|  |  |
| --- | --- |
| |  | | --- | | 비디오 게임이 대중화된 이후에도 TRPG는 멸종하지 않고 꾸준히 사랑받아왔습니다 물론 현재도 많은 TRPG게임이 나오고 있죠.(창세기전, 천량열전 등등) 비디오 게임이 시각적인 효과는 뛰어났지만 혼자서 즐겨야 하는 반면, TRPG는 사용자의 상상에 의존해야 하지만 다수가 함께 즐길 수 있다는 장점이 있습니다. 이러한 경향은 [머드게임](http://dic.search.naver.com/search.naver?sm=ncc_clk&where=kdic&query=%B8%D3%B5%E5%B0%D4%C0%D3&x=31&y=9)(Multi-User Dungeon)으로 이어지게 됩니다. 머드게임이란 TRPG를 온라인으로 옮긴 게임으로, GM의 역할은 서버가 맡고 사용자간 음성 대화는 자판으로 입력하는 텍스트가 대신했하는 혁신적인 장르 였습니다. 또한 온라인의 특성상 TRPG보다 훨씬 더 많은 사용자들이 함께 즐길 수 있었기도 하구요. 머드게임은 1990년대 후반까지 PC통신 사용자들에게 큰 인기를 끌었습니다.  이후 머드게임에 그래픽 효과를 추가한 그래픽 머드게임(graphical MUD, 혹은 [머그게임](http://dic.search.naver.com/search.naver?sm=ncc_clk&where=kdic&query=%B8%D3%B1%D7%B0%D4%C0%D3&x=28&y=21))이 등장했고, 이 그래픽 머드게임이 바로 초창기 MMORPG에 해당합니다. 대한민국 국민이라면 (-\_-'') 모르는 사람이 없는 국산게임 ‘[바람의 나라](http://dic.search.naver.com/search.naver?sm=ncc_clk&where=kdic&query=%B9%D9%B6%F7%C0%C7+%B3%AA%B6%F3&x=20&y=26)’를 비롯해 ‘에버퀘스트’, ‘다크 에이지 카멜롯’, ‘[울티마 온라인](http://dic.search.naver.com/search.naver?sm=ncc_clk&where=kdic&query=%BF%EF%C6%BC%B8%B6+%BF%C2%B6%F3%C0%CE&x=10&y=22)’ 등 현재 1세대 MMORPG로 꼽히는 게임들도 초기에는 모두 그래픽 머드게임으로 불렸네요. 이 중 ‘울티마 온라인’의 개발자 [리처드 게리엇](http://search.naver.com/search.naver?where=nexearch&sm=ncc_clk&ie=utf8&query=%EB%A6%AC%EC%B2%98%EB%93%9C%20%EA%B0%9C%EB%A6%AC%EC%97%87)  Richard Garriott, 1961~)이 MMORPG라는 신조어를 만들어 냈고, ‘울티마 온라인’의 인기에 힘입어 MMORPG라는 용어도 대중들에게 알려지게 됐습니다.  국내에서는 넥슨의 ‘바람의 나라’, 엔씨소프트의 ‘[리니지](http://dic.search.naver.com/search.naver?sm=ncc_clk&where=kdic&query=%B8%AE%B4%CF%C1%F6&x=17&y=29)’가 큰 인기를 끌면서 수많은 MMORPG가 시장에 선보이기 시작했습니다. 현재 MMORPG는 FPS게임, RTS게임과 더불어 한국에서 가장 사랑 받는 온라인 게임 장르로 자리잡았고, 아직도 많은 mmorpg 게임이 생산되고 있습니다. | |



대표적인 MMORPG 게임의 하나인 ‘리니지’

|  |  |
| --- | --- |
| |  | | --- | | **최초의 MMORPG는 무엇인가?**  이처럼 MMORPG는 그래픽 머드게임과 딱히 구분짓기 어렵습니다. 따라서 최초의 MMORPG가 무엇인가에 대해 의견이 분분한데요. ‘아일랜드 오브 케스마이(Island of Kesmai)’가 최초라고 여기는 사람도 있고, ‘네버윈터 나이츠(Neverwinter Nights)’나 ‘울티마 온라인’이 최초라고 믿는 사람도 있습니다. 또한 넥슨도 ‘바람의 나라’가 최초의 MMORPG라고 주장하고 있습니다. (자기꺼니까.....)  1985년 등장한 ‘아일랜드 오브 케스마이’는 아스키 그래픽을 적용한 최초의 게임인데요.. 기존 문자를 활용해 그림을 그리는 방식으로, 각 문자마다 특정한 의미를 담고 있습니다.. 예를 들면 ‘-‘와 ‘I’를 여러 개 이으면 벽이 되고 [‘@’](mailto:%E2%80%98@%E2%80%99)는 사용자 자신을, ‘+’는 문을 뜻 하는데요.. 아래 자료사진처럼 정말 어지럽습니다...... 그래픽보다는 문자에 가깝기 때문에 그래픽 머드게임으로 분류하지는 않습니다.  1991년 등장한 ‘네버윈터 나이츠’는 최초로 ‘진짜’ 그래픽을 기반으로 한 게임인데요. 이 게임은 시간당 6달러에 달하는 비싼 요금에도 불구하고 1997년 서비스를 종료할 때까지 누적사용자가 11만5000명에 달할 정도로 인기를 끌었스빈다 인터넷이 보편화 되지 않았던 시절이었던 것을 감안한다면 아주 놀라운 기록인 것 같네요..(.사실 게임에 관심이 없으신분들은 대단하지 않은 수로 여길 수도 있으나...) | |

|  |  |
| --- | --- |
| http://ncc.phinf.naver.net/ncc02/2011/6/15/108/img_04-1.jpg  아스키 그래픽을 사용한 게임 | http://ncc.phinf.naver.net/ncc01/2011/6/15/180/img_04-2.jpg  네버윈터 나이츠(출처: <http://www.mobygames.com/)> |

|  |  |
| --- | --- |
| |  | | --- | | 1996년에는 국내 최초의 그래픽 머드게임 ‘바람의 나라’가 등장했습니다!!!!!!!/(개인적으로 가장 좋아하고 현재도 즐기고있는 게임인데요(ㅎㅎㅎ). 이 게임은 텍스트 기반의 머드게임에 아쉬움을 느끼던 국내 사용자들에게 폭발적인 인기를 얻었고, 이후 등장한 수많은 국내 MMORPG에 지대한 영향을 미쳤습니다. 2005년에 무료 게임으로 전환했으며, 지금도 꾸준한 인기를 누리고 있습니다. 개인적인 여담이지만 그전까지는 유료게임 이었기때문에 레벨20짜리 캐릭터가 무수히 많았죠...(레벨20까지만 무료, 그이상은 유료)  ‘바람의 나라’보다 조금 늦은 1997년, ‘울티마 온라인’이 정식 서비스를 실시했습니다. MMORPG라는 용어가 이 때 처음 등장했기 때문에, ‘울티마 온라인’을 기준으로 이전 게임은 그래픽 머드게임, 이후 게임은 MMORPG로 구분하는 사람들도 있습니다. | |

|  |  |
| --- | --- |
| http://ncc.phinf.naver.net/ncc02/2011/6/15/171/img_05-1.jpg  국내최초의 그래픽 머드게임 ‘바람의 나라’ | http://ncc.phinf.naver.net/ncc01/2011/6/15/119/img_05-2.jpg  MMORPG라는 이름을 처음 쓴 ‘울티마 온라인’ |

|  |  |
| --- | --- |
| |  | | --- | | **MMORPG의 주요 특징**  **캐릭터의 성장**  MMORPG는 RPG의 일종이기 때문에, RPG의 기본 특징을 모두 보유하고 있다. 그 중 하나가 RPG의 목적이기도 한 ‘캐릭터 성장’입니다.. 사용자가 임무를 수행하거나 몬스터를 잡으면 경험치 포인트(experience points)를 획득하게 되고, 이 경험치 포인트가 일정 수 모이면 캐릭터의 ‘레벨’이 오르게 되는데요, 어려운 임무(퀘스트)를 맡거나 강한 몬스터를 잡을 경우 그에 합당한 보상(예를 들면 레어무기, 레어방어구)이 따라오는데, 이 보상을 통해 캐릭터의 능력치를 올립니다.  **역할 수행**  RPG의 기본 특징 중 하나는 다양한 직업을 제공한다는 점이빈다. 예를 들어 사용자가 ‘마법사’라는 직업을 선택하면 게임 안에서 해당 직업의 역할을 수행하게 되고, 또한 마법사에 어울리는 방어구를 착용해야 하며, 마법사만이 쓸 수 있는 기술을 배우게 됩니다.  **상호 작용**  MMORPG의 배경은 일종의 가상 세계입니다. 따라서 현실과 마찬가지로 다른 사람들과 의사소통을 하지 않으면 게임을 즐길 수 없죠. 기본적으로 대화를 주고받는 채팅 시스템을 지원하며, 친목을 다질 수 있는 커뮤니티 시스템인 ‘길드’ 또는 ‘클랜’을 구성할 수 있다는 것이 또다른 특징입니다. 전투를 벌일 때도 각자 역할을 분담해 수행하게 되며, 사용자간 필요한 물건을 교환하거나 사고 팔 수도 있습니다.  **시간의 연속성**  혼자서 즐기는 게임은 게임 속의 시간을 조절할 수 있습니다. 사용자가 접속을 종료하면 게임 속 시간은 멈추고, 게임을 재개하면 다시 시간이 흐른다. 하지만 MMORPG의 시간은 실제 시간과 똑같죠........ 심지어 서버에 아무도 접속하지 않은 상황에서도 게임 세계의 시간은 계속 흐르게 됩니다. 사용자의 모든 정보는 기본적으로 저장되지만, 혼자서 즐기는 게임처럼 특정 과거 시점으로 되돌아가지는 못합니다.  **MORPG와의 차이**  MMORPG와 [MORPG](http://dic.search.naver.com/search.naver?where=kdic&sm=ncc_clk&ie=utf8&query=MORPG)를 혼동하는 사람들도 있는데요, MORPG는 MMORPG에서 대규모를 뜻하는 ‘Massive’를 뺀 게임입니다. 따라서 접속자의 수에 따라 MMORPG와 MORPG의 차이가 드러나게 되는데요, 일반적으로 ‘방’을 만들어 그 방에 접속한 사람들끼리만 전투 및 사냥을 한다면 MORPG, 모든 사용자들의 왕래가 가능한 공개 지역에서 전투 및 사냥을 한다면 MMORPG로 구분짓습니다. MORPG 게임의 사례로는  '던전앤파이터', '**디아블로♥♥♥♥**', 'C9' 등이 있습니다. 하지만 최근에는 서로의 특징을 흡수한 게임들이 등장하면서 MORPG와 MMORPG의 구분이 어려워지고 있습니다. | |

대표적인 MORPG 게임 중 하나인 ‘던전앤파이터’

**[출처]** [MMORPG란 무엇인가?.](http://blog.naver.com/uiiam/150135755746)|**작성자** [아임](http://blog.naver.com/uiiam)

==============================================================================

**Ⅲ**

접한 질문 이지만 답변을 해주시면 감사하겠습니다.

저는 아주 간단한 온라인 게임을 만들어 보긴 했는데요.

이런 게임 종류는 처음 이라서 말이죠.

저는 서버를 다음과 같이:

1. 게임 서버 (더 작게 나누자면: 아이템 서버, 몬스터 서버, 유저 서버, NPC 서버로 나뉨)

2. 로그인 서버;

이렇게 크게 두가지로 나누구요.

서버가 맨처음 시작되면 데이터베이스에 저장된 초기화 정보들을 "모두" 메모리에 로딩시키고 메모리에서 업데이트가 이루어 집니다.

물론 중요한 정보 (예를 들면 아이템의 소유권) 는 바로 데이터 베이스에 업데이트를 해줍니다.

하지만 중요하지 않은 정보 (예를 들면 몬스터의 좌표같은것) 는 메모리 에서만 업데이트를 합니다.

클라이언트는 게임에 접속하면 자신에게 필요한 정보(예를 들자면: 유저 좌표 주위에 아이템, 몬스터 정보들)을

서버가 메모리에서 찾아서 패킷으로 넘겨주고, 클라이언트는 그 정보를 토대로 화면을 출력합니다.

아직 구현은 안하고 상상만으로 종이에 그려보고 있습니다.

저는 게임 프로그래밍을 전공으로 한번도 배우지 않아서 엉터리일 확률이 100% 입니다.

웃지마시기를...ㅠㅠ 으허헐헣허허헝ㄴ

여기서 질문인데요.

1. lineage 같은 mmorpg 의 기본 서버 구조와 구성은? 간략하게 설명혹은 링크 부탁드립니다.

2. 실시간으로 바뀌는 게임의 정보(몬스터, 아이템, 유저등...) 는 지금 내가 하듯 memory 에 모두 로딩하여 관리 하는가?

아니면 sql DB 에 직접 수정을 가하는가? 혹은 적절한 조합?

3. 클라이언트에게 정보(패킷)을 보낼때, 객체(혹은 스트럭쳐)단위로 보내야 하는가?

혹은 데이터 타입 단위로 쪼개어 보내야 하는가?

혹은 스트링으로 한줄에 보내는가?

혹은 더욱 좋은 방법이 존재하는가?

4. 이와 같은 서버를 만들때 언어의 중요성과 운영체제의 중요성은?

추천하는 운영체제와 언어는 무엇인가?

답변을 기다리겟습니다.

1. 리니지와 같은 대형 게임의 경우에는 질문자님이 말씀하신것과 비슷하게 나뉘어있습니다. 실제 리니지의 게임서버 구성은 대외비에 해당될테니 알기는 쉽지 않지만 로그인서버, 게임서버, 몬스터서버, 인던서버 등으로 나뉘게 됩니다.

2. 기본적으로 서버 시작후 데이터 입출력은 메모리에서 한다고 생각하셔야 합니다. 중간중간 중요한 데이터들은 데이터베이스에 바로 기록을 해야하겠지만 그렇지 않은 데이터(서버가 죽어서 날아가도 게임상 전혀 지장이 없는 데이터)는 메모리에서 입출력을 해야합니다. 데이터베이스를 임시 데이터 기록 용도로 사용하실 생각은 하면 안됩니다. 디스크 io보다 더 느린게 데이터베이스 입출력일수도 있습니다.

그래서 데이터베이스를 게임서버에서 직접 엑세스 하는게 아닌 dbms서버 앞단에 간단한 디비캐시 서버를 두어  
사용하는 경우도 있습니다만 그렇다고 해서 데이터베이스를 임시 데이터 기록 용도로 사용해선 안됩니다.

3. 데이터를 보내는 방법은 각각 장단점이 있으니 어떤게 좋다고는 말씀드리기가 힘듭니다.  
크게 두가지로 나누자면 바이너리와 텍스트 방식이 있습니다.

바이너리는 메모리에 있는 데이터를 그대로 전송하는 것이고 텍스트는 메모리의 데이터를 텍스트형식으로  
인코딩하여 전송하는 방식입니다.

바이너리의 경우에는 클라이언트와 서버의 cpu가 동일한 엔디언이라면 상관이 없지만 서로 다른 엔디언 방식을  
사용한다면 이를 조절해서 전송해줘야합니다.

텍스트의 경우에는 엔디언을 신경쓰지 않아도 되지만 전송시 인코딩과 받는쪽에서 디코딩과정이 필요하므로 추가적인 연산이 필요합니다. 그리고 바이너리에 비해서 패킷 분석이 쉬우므로(암호화 하면 달라집니다만) 해킹에 위험합니다.

일반적으로 바이너리로 보낼때는 구조체에 넣어서 전송을 하게 됩니다. 데이터타입별로 하나씩 보낼려면 귀찮죠.. 한꺼번에 구조체에 보낼것을 몰아넣은 다음에 구조체 크기만큼 전송을 하면 간편하니까요..

4. 언어는 대부분 c/c++을 이용하여 만듭니다. 아직까진 그래도 c/c++만큼 성능 나오는 언어가 없기 때문입니다.  
운영체제는 예전에는 비윈도우계열을 많이 썼지만 요즘은 점점 윈도우계열 서버를 사용하는게 늘어난다고 합니다. 일단 개발이 편하기도 하고 운영체제에 문제가 생겨 서비스가 중단되었을 경우 ms에서 손해배상도 해준다고 하네요..(손해배상 부분은 들은 이야기라 확실하진 않습니다.)