

리얼 라이브 엔진 *Step by Step Manual* By 마이아크

일단 찾는 법 자체는 착한님의 "리얼 라이브 초간편 코드 찾기 메뉴얼" http://www.aralgood.com/zbxe/board_lecture/75386을 베끼다 싶이 했습니다. 예제만 다르죠.

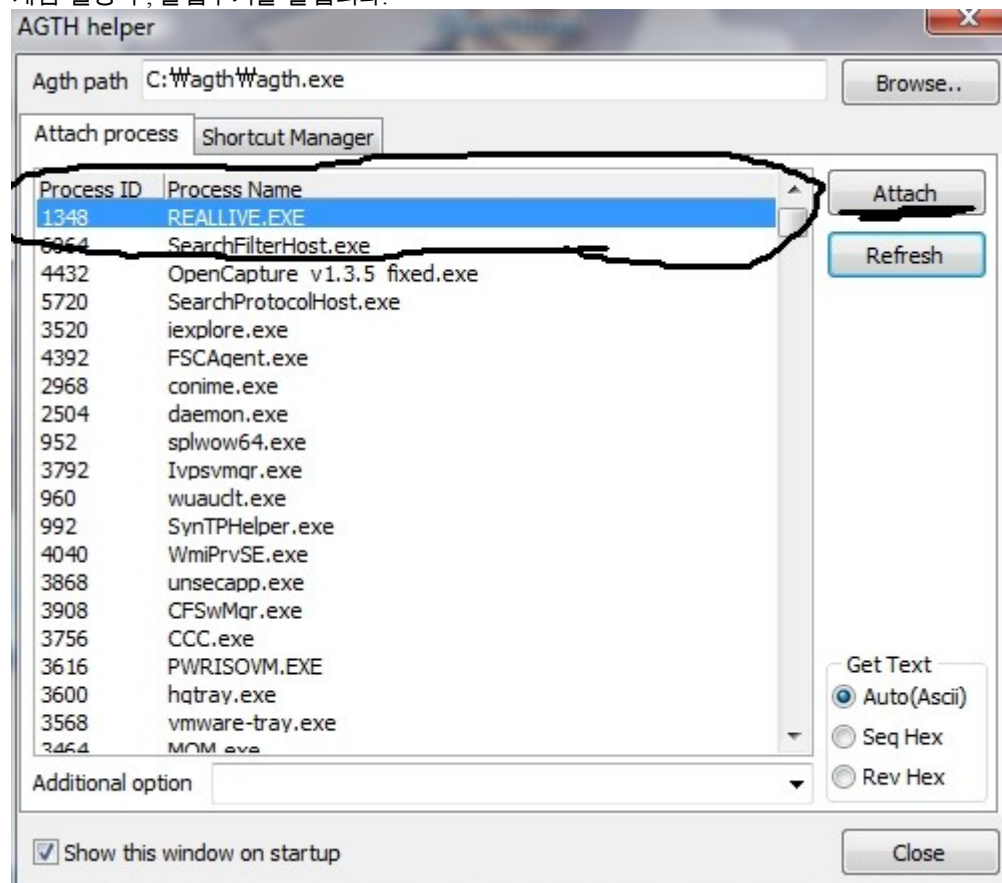
준비사항:

1. 게임을 설치 할 수 있는 지식.
2. 리얼라이브 엔진 게임을 구별할 수 있는 지식.
3. 어플로케일
4. 최신 아라트랜스
5. EditPlus2
6. UltraEdit
7. Ollydbg 1.10
8. Cliphooker & 최신 AGTH

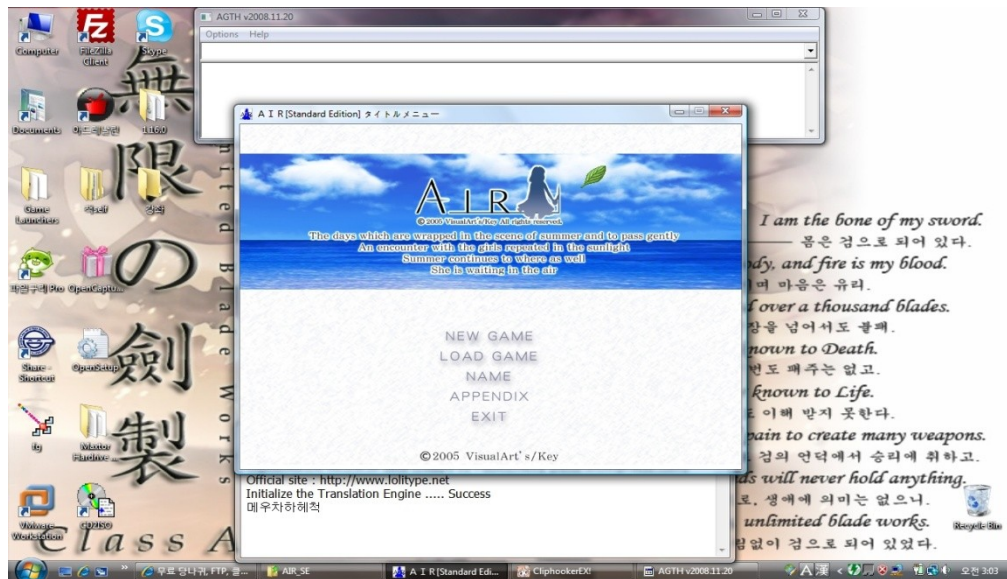
자, 이제 찾아보죠.

예제는 AIR Standard Edition.
게임을 어플로 설치 후, 어플로 실행합니다.

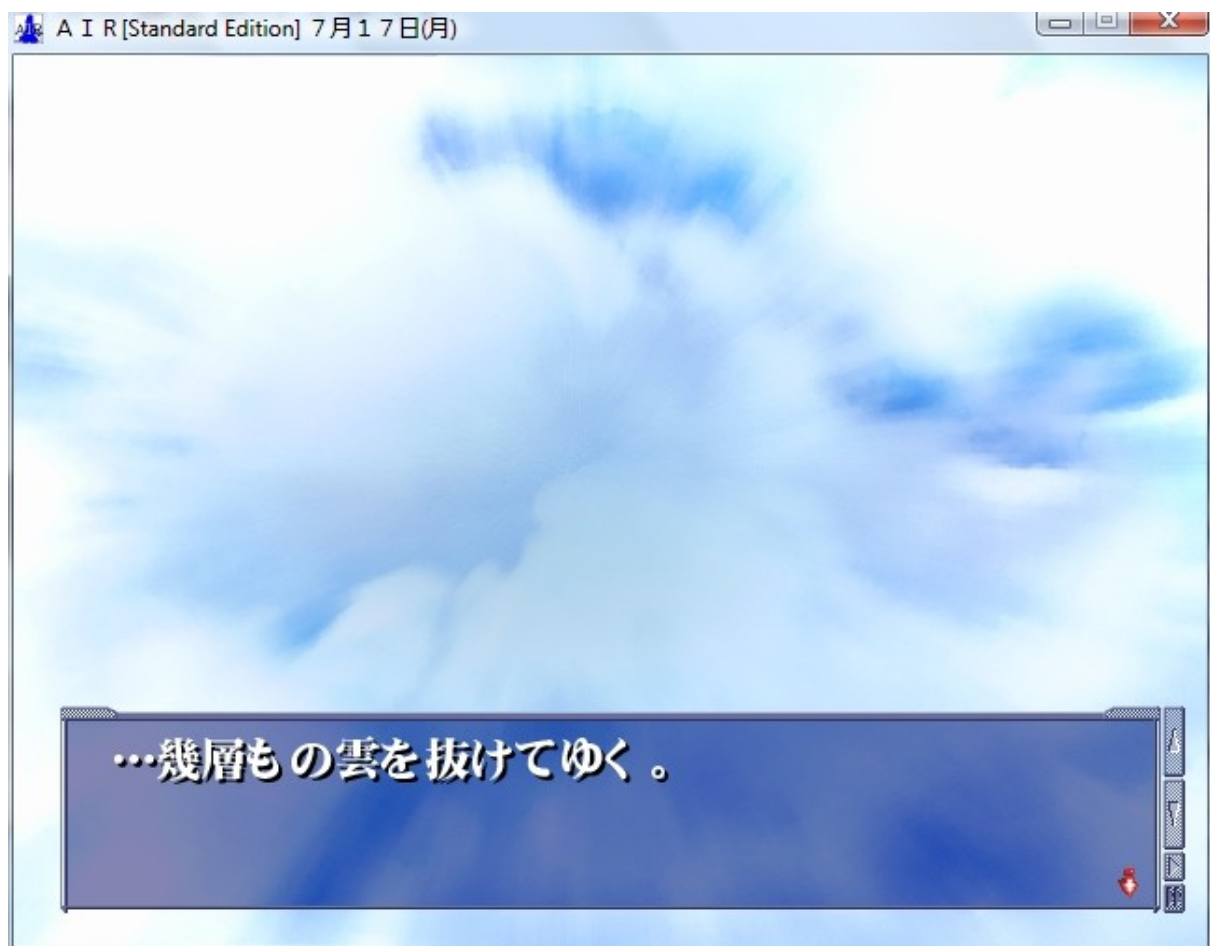
게임 실행 후, 클립후커를 돌립니다.



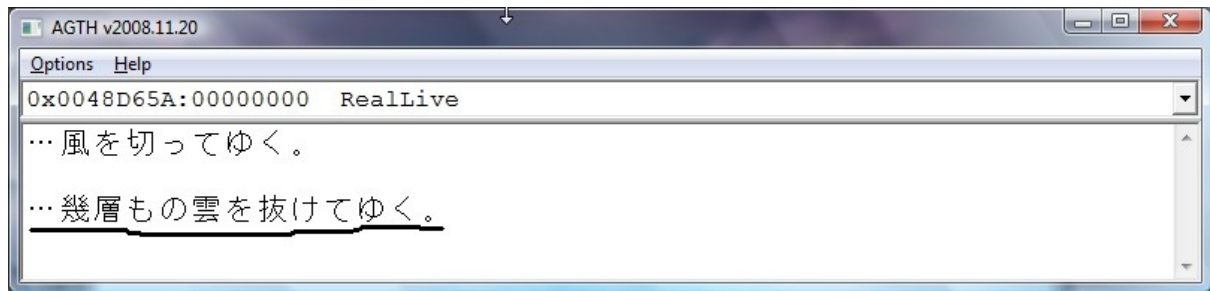
REALLIVE.EXE 가 게임 클릭 후, Attach 버튼을 누르면.....



성공!!!!
이제 게임을 실행합니다.

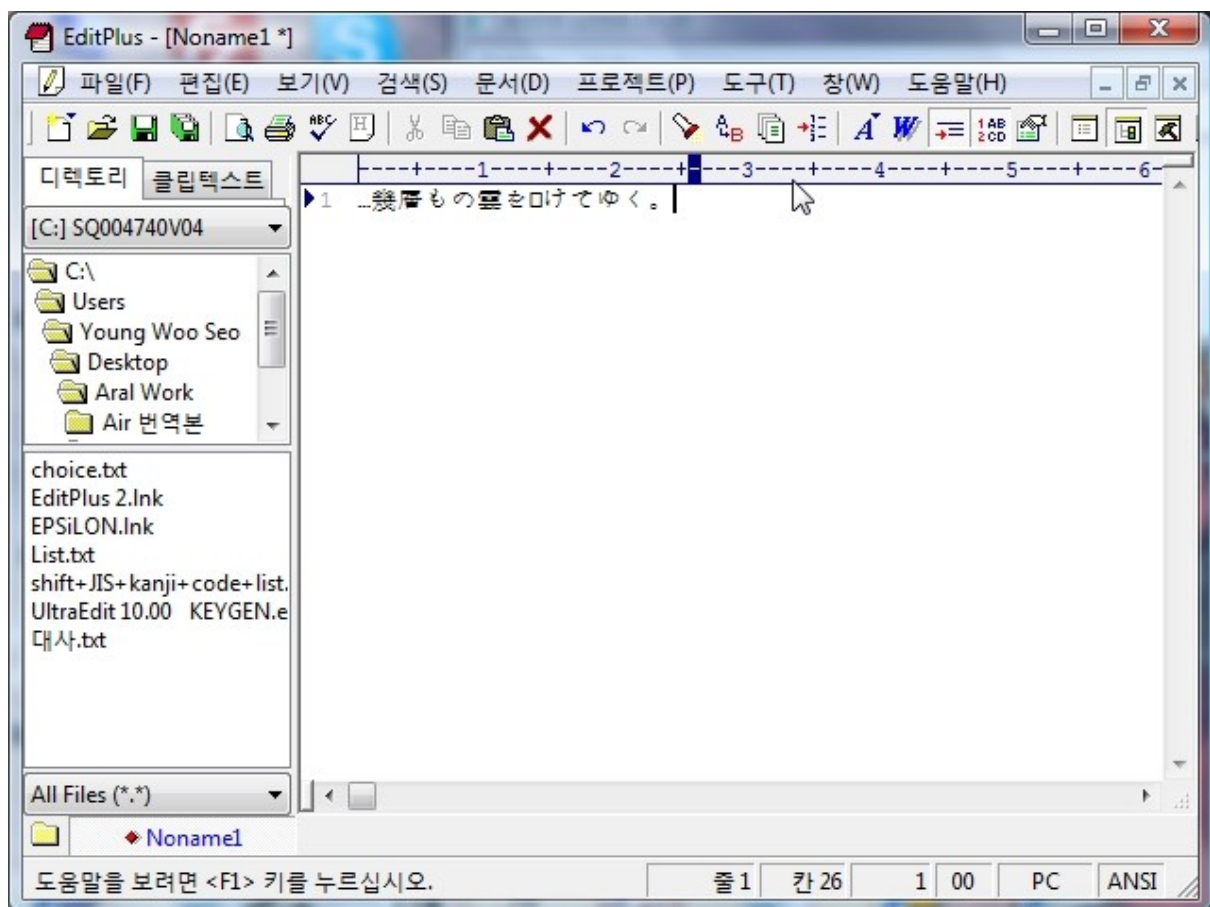


첫번째대사는 났고....두번째까지 진행합니다. (윗 스샷은 두번째 대사)



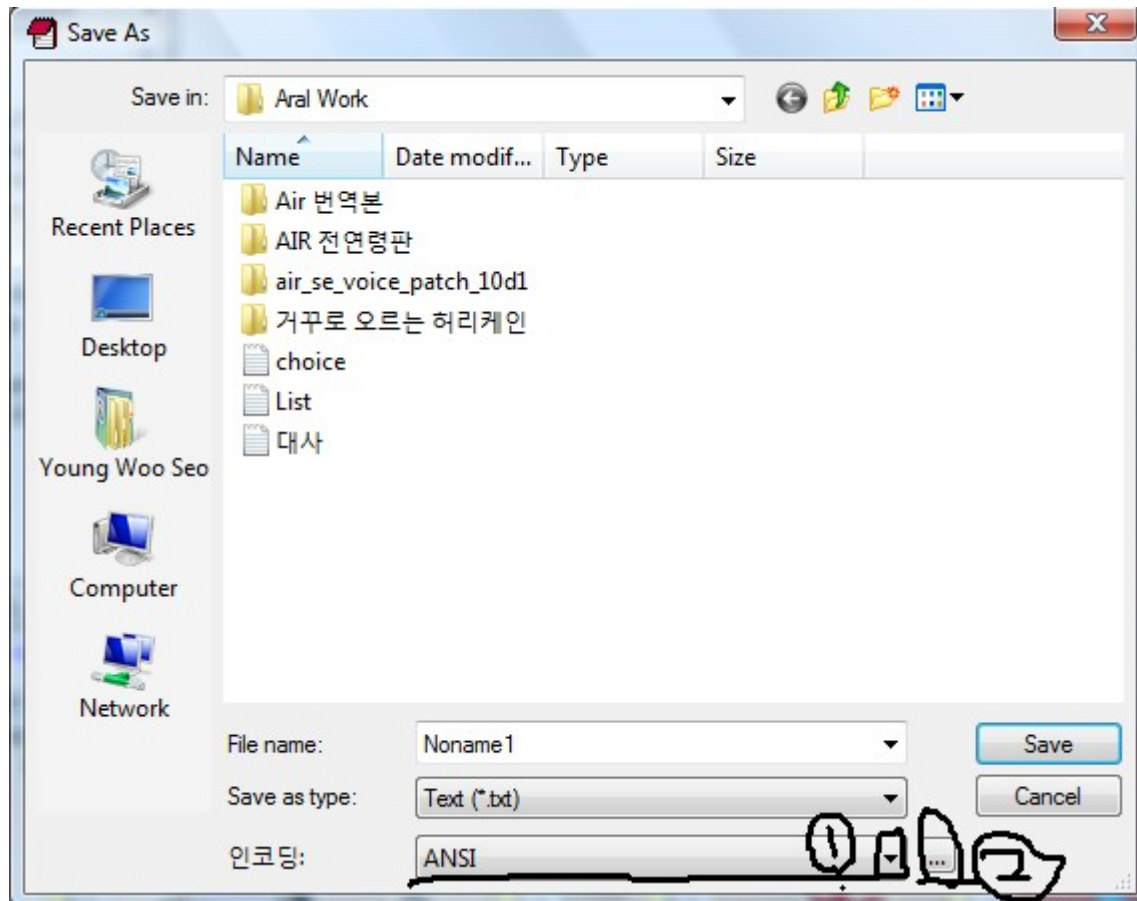
저 두번째 대사를 복사합니다. (CTRL+C)

그리고.....



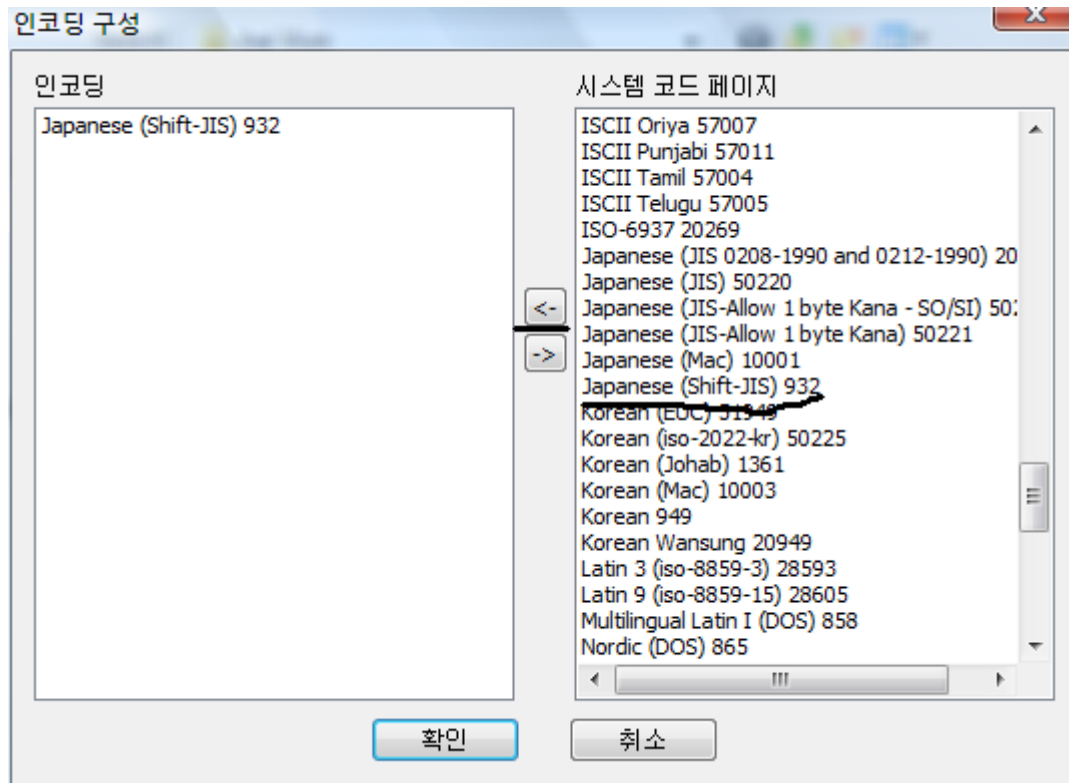
이렇게 붙여넣기를 합니다.

이제 저장을 합니다.



인코딩을 Japanese(Shift-JIS)932 로 바꿔줍니다.

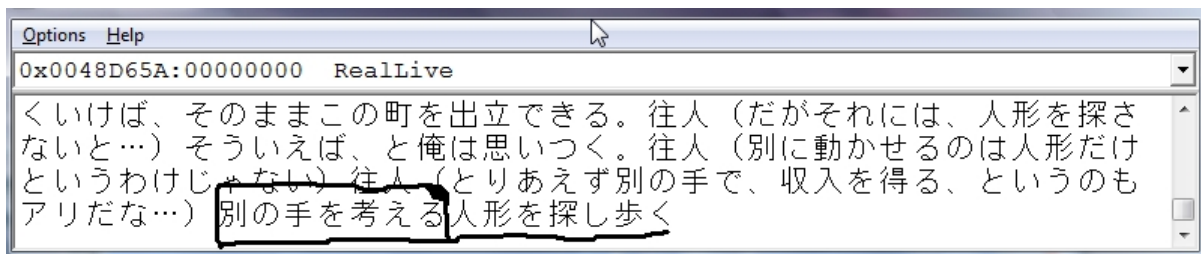
만약에 그러한 인코딩이 없다 하시면 2 번을 눌러주세요 ^^



이러한 창이 뜹니다. 스크롤 다운 후, Japanese (Shift-JIS) 932 을 클릭 후 오른쪽으로 옮겨줍니다.
자, 이제 저장합니다.



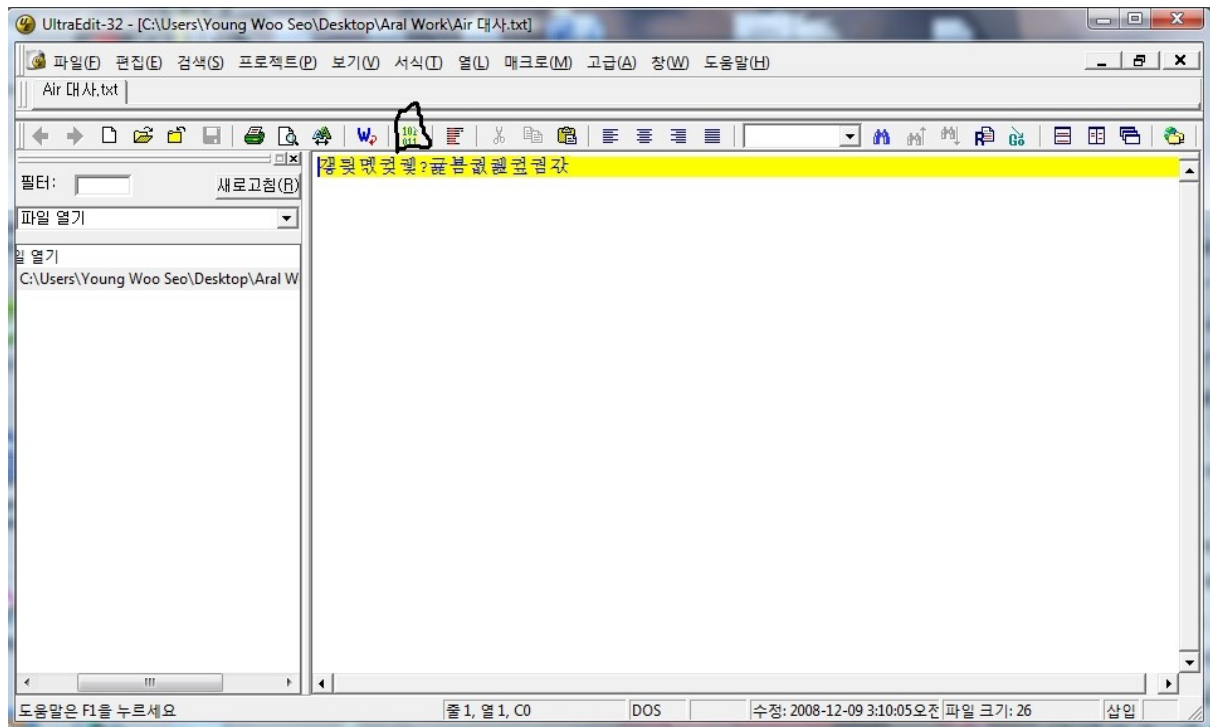
햇김에 선택지까지 수집하죠. 대사와 마찬가지로 이 상태에서 수집합니다.



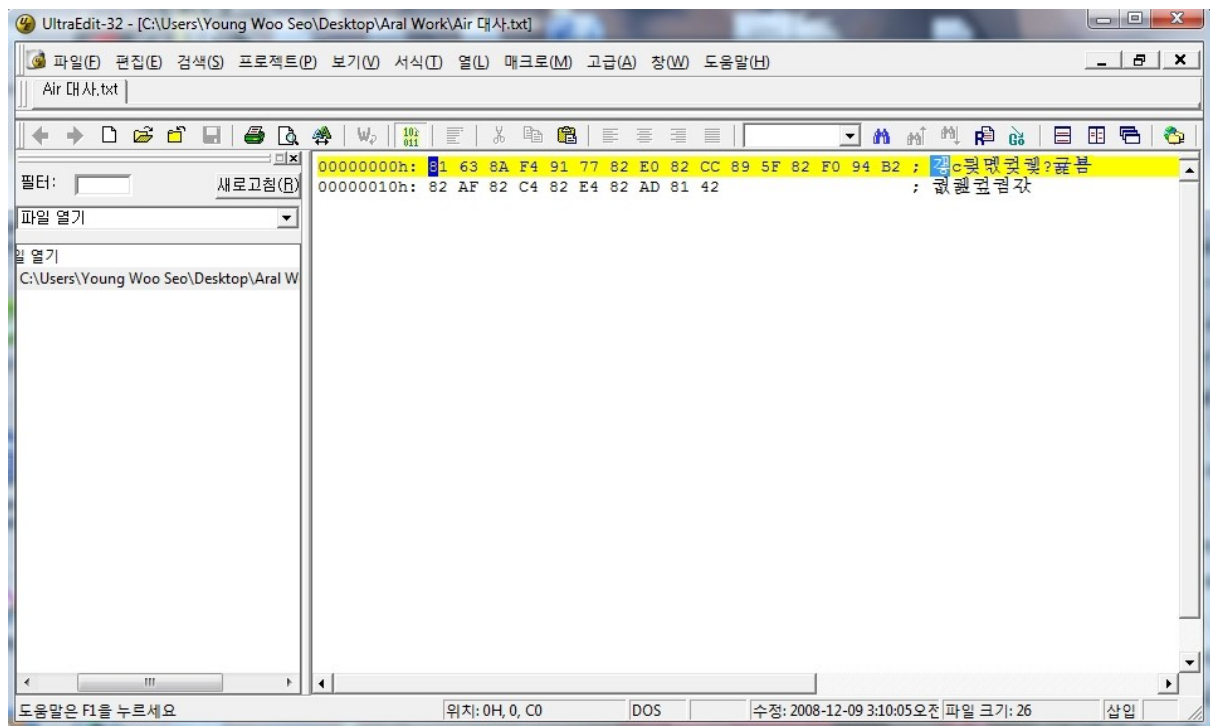
사각형과 밑 줄이 두 개의 선택지입니다. 첫 번째 선택지만 수집하죠. (AGTH의 상태가 저런건 CTRL 스킵신공으로 인해..)

대사와 선택지 수집이 끝났습니다.

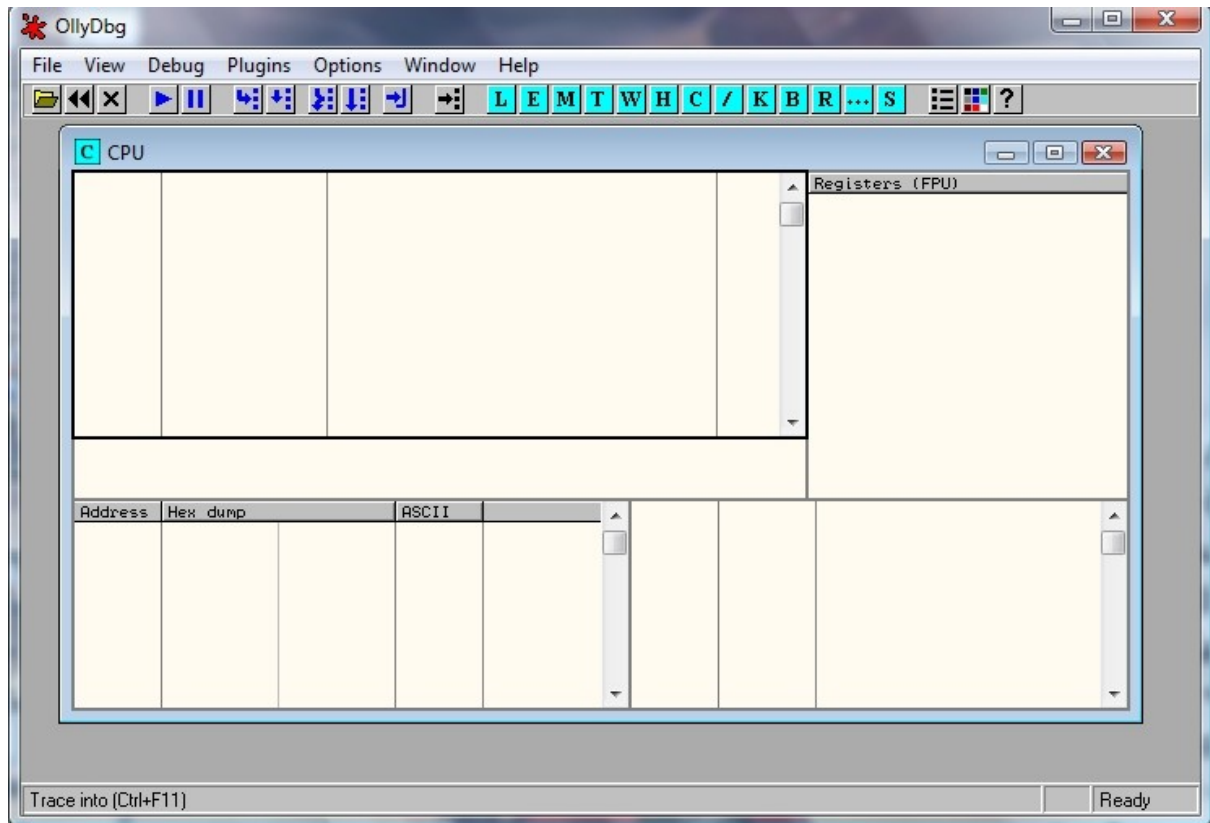
이제 울트라 에디트를 킵시다.



대사를 불러오면 요렇게 뜹니다. 동그라미(?) 쳐진 곳을 클릭해 16 진수를 보이게 합시다.



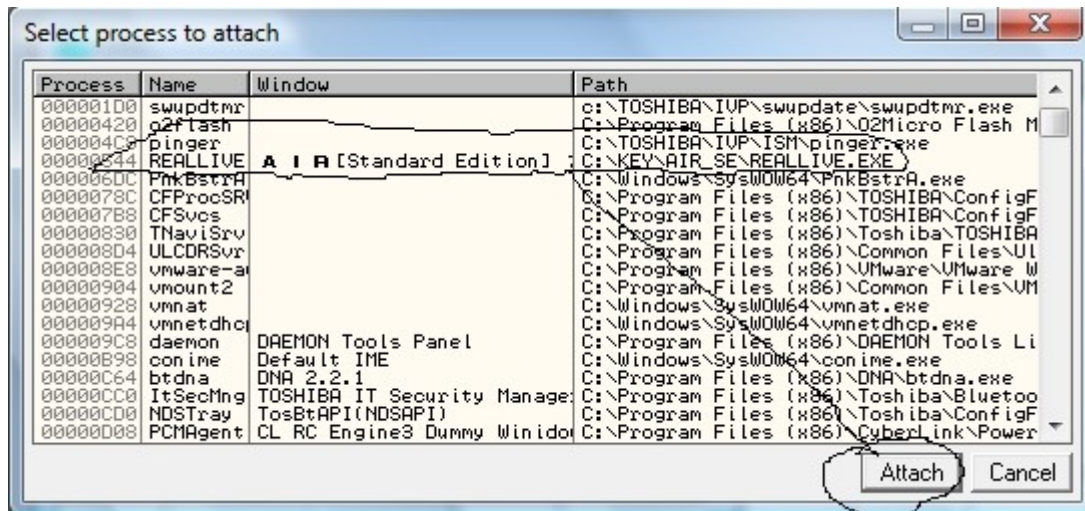
성공!! ^^
게임타이틀로 돌아갑니다.
이제 올리를 킵시다.



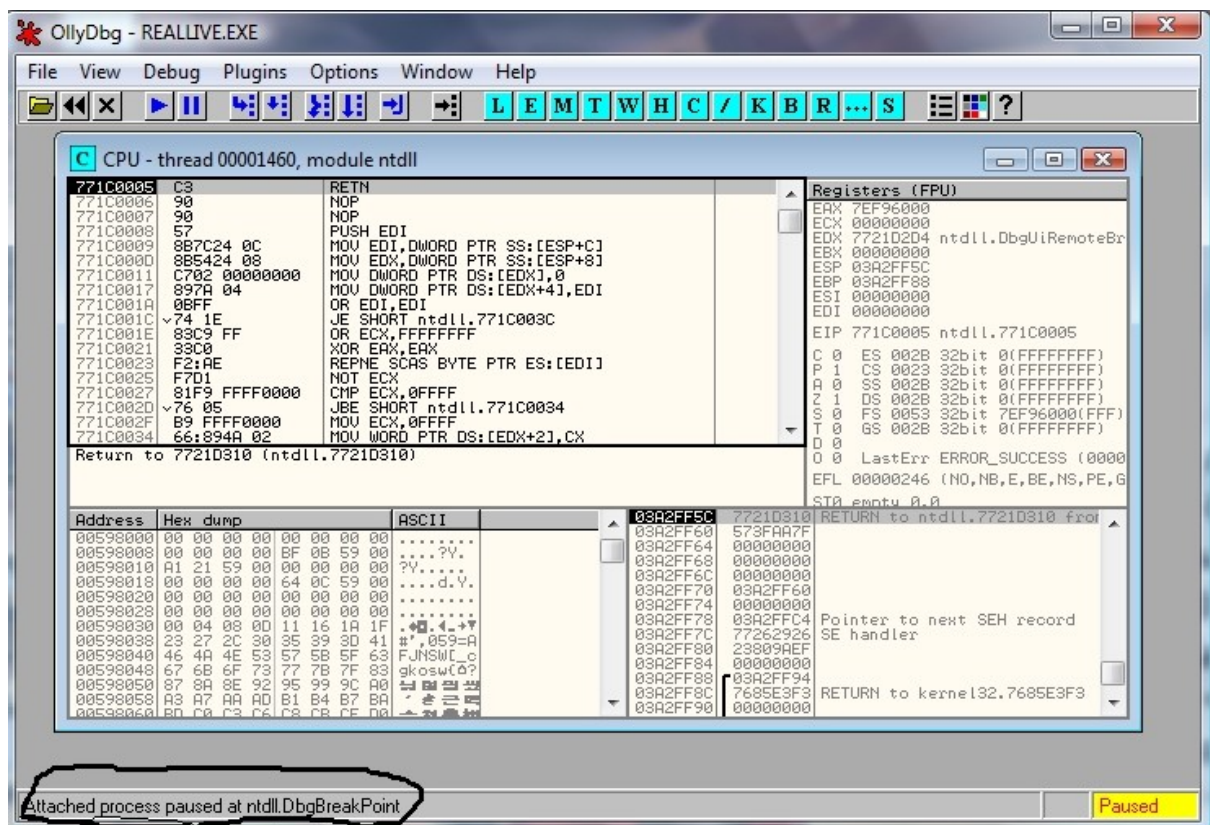
File 메뉴에 들어가 게임을 Attach 시킵시다.



이렇게.....

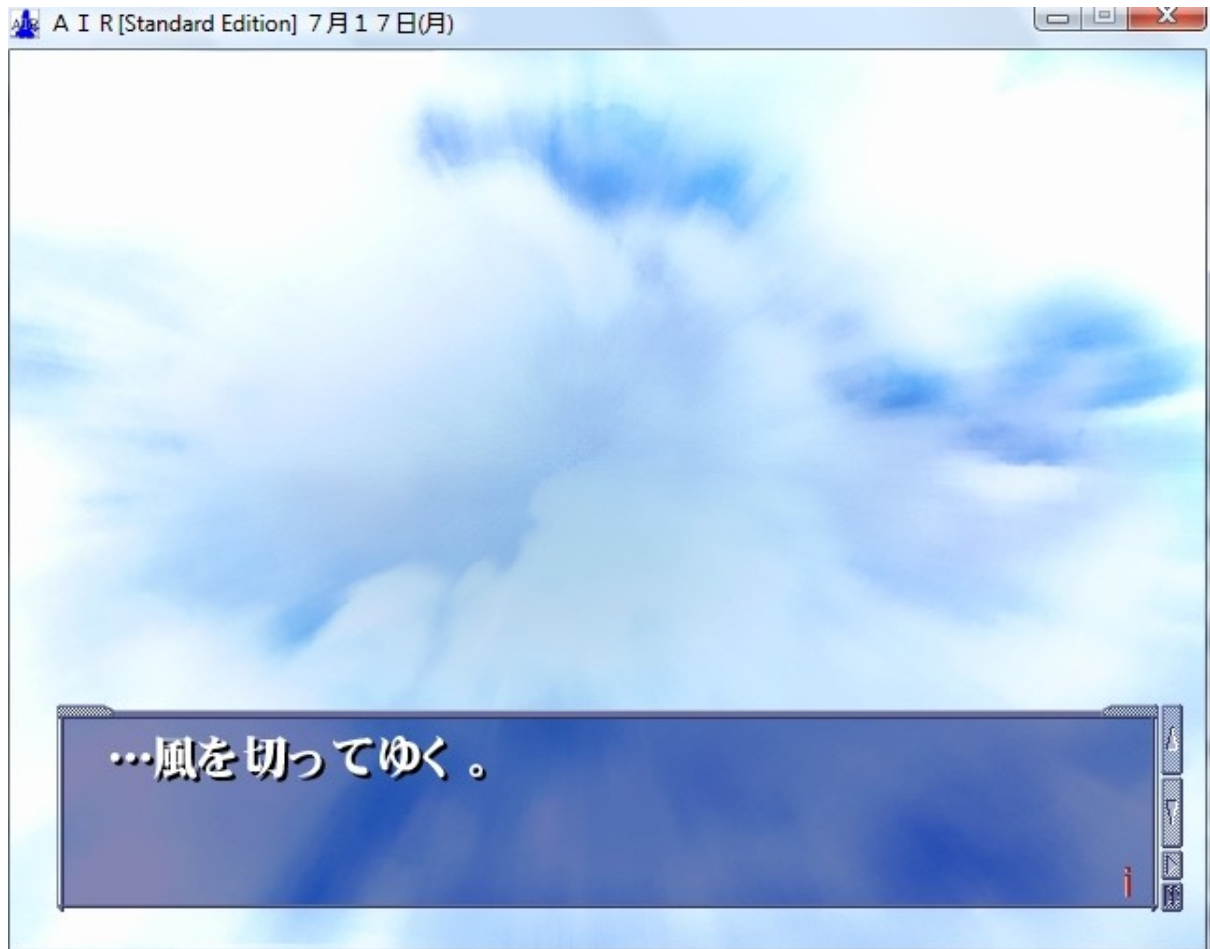


자 Attach 시킵시다.



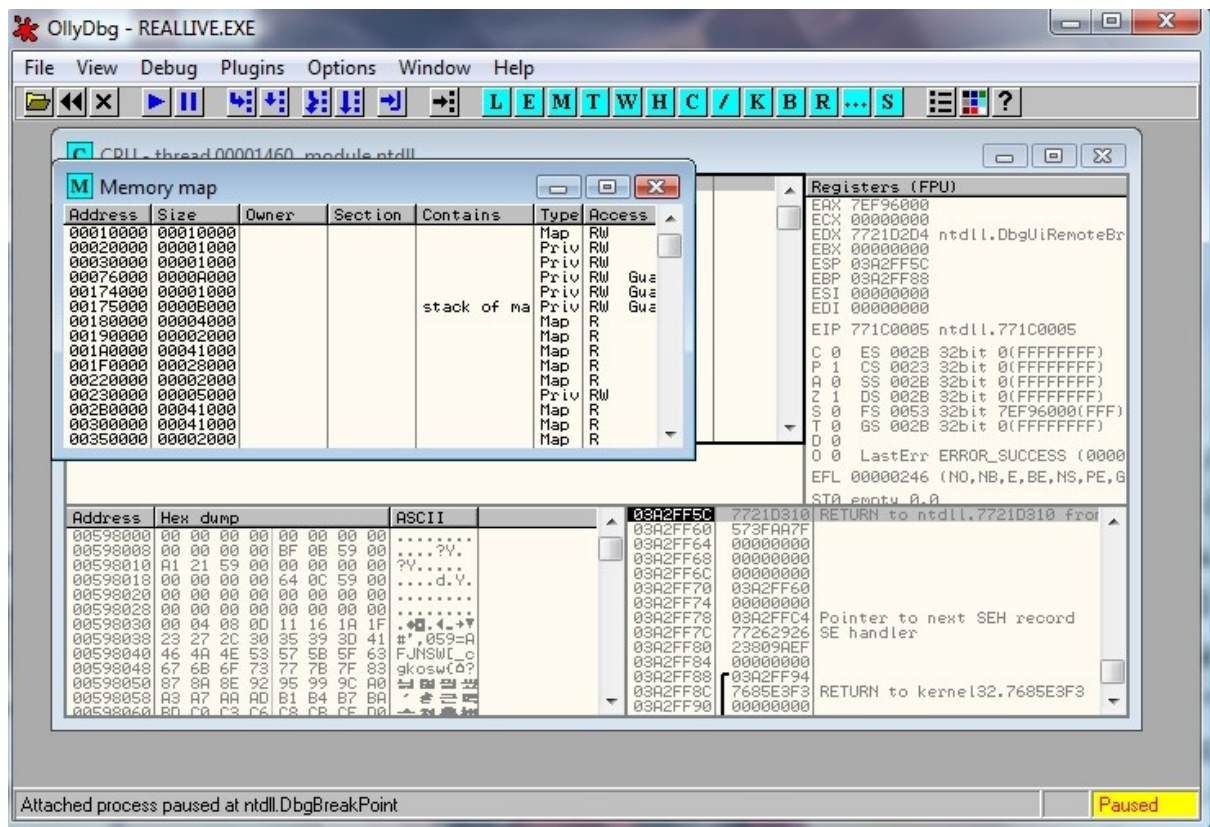
이 부분에서 멈출 겁니다. 살포시~F9 을 눌러줍시다.

그 후, 게임을 수집한 대사 전!까지 진행합시다.

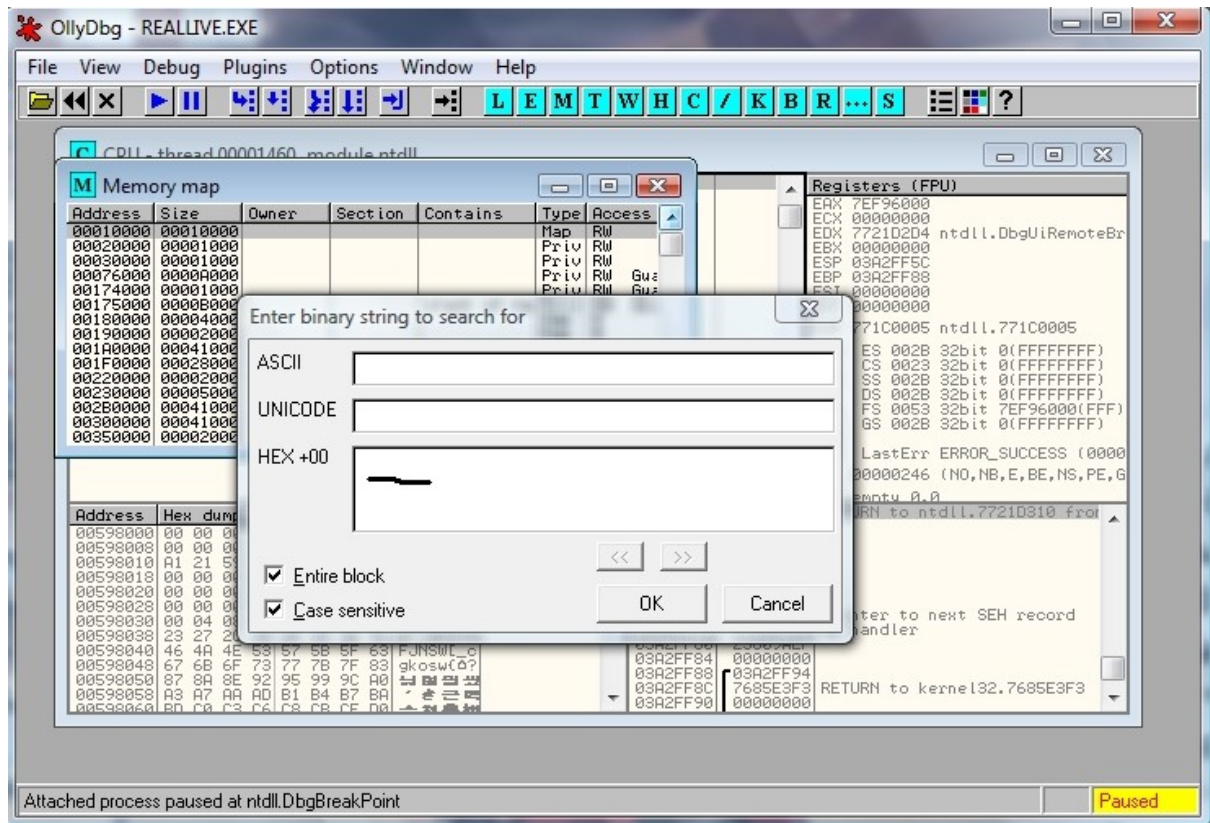


이제 본격적인 디버깅을 합시다.

위에 있는 올리 스샷에서 Alt+M 을 누르시면...



Memory Map 이 뜹니다. 그 후, CTRL+B 를 누르시면...



밑 줄 친곳에 울트라 에디트에 나와있는 16 진수를 써놓습니다.

The screenshot shows the OllyDbg interface with the following components:

- Menu Bar:** File, View, Debug, Plugins, Options, Window, Help.
- Toolbar:** Includes icons for file operations, navigation, and search.
- Memory Map (Left Panel):** Displays memory addresses, sizes, owners, sections, and access permissions. The address range shown is from 00010000 to 00030000.
- Registers (Right Panel):** Displays the current state of CPU registers (EAX, ECX, EDI, etc.) and their values.
- Search Dialog (Center):** A modal dialog titled "Enter binary string to search for" with three input fields:
 - ASCII:** Contains the string "c ô w à î _".
 - Unicode:** Contains the string "E 1".
 - HEX +OC:** Contains the hex string "81 63 8A F4 91 77 82 E0 82 CC 89 5F".
 The dialog also includes checkboxes for "Entire block" and "Case sensitive", and buttons for "<<", ">>", "OK", and "Cancel".
- Disassembly (Bottom Panel):** Shows the current instruction being executed, which is a "RETURN to kernel32.7685E3F3".
- Status Bar:** Displays "Attached process paused at ntdll.DbgBreakPoint" and "Paused".

The screenshot shows the OllyDbg interface with the following components:

- File View Debug Plugins Options Window Help** menu bar.
- Memory map** window showing a table of memory segments:

Address	Size	Owner	Section	Contains	Type	Access
030BD000	00001000				Priv	RW
030BE000	00002000				Priv	RW
030DF000	00100000				Priv	RW
043FD000						
043FE000						
044FD000						
044FE000						
045FD000						
045FF000						
046FD000						
046FF000						
04700000						
04830000						
04922000						

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Dump - 04830000..0488CFFF** window showing a hex dump of memory:

```

04830000 81 63 8A F4 91 77 82 E0 82 CC 89 5F 82 F0 94 82 ...
04830001 82 AF 82 C4 82 E4 82 AD 81 42 23 00 03 11 00 00 ...
04830002 82 00 8A 3F 82 10 82 00 82 C7 82 81 82 DC 82 05 ...
04830003 82 E0 81 41 82 C7 82 B1 82 DC 82 C5 82 E0 80 82 ...
04830004 82 DD 82 D6 81 42 23 00 03 11 00 00 00 0A 40 ...
04830005 00 00 04 00 8B B9 82 CC 93 AE 9C A7 82 AA 91 81 ...
04830006 82 A2 81 42 23 00 03 11 00 00 00 0A 41 00 40 ...
04830007 05 00 91 C1 82 C0 82 E0 82 AA 81 41 95 F6 82 EA ...
04830008 37 8E 82 8F 82 BB 82 A4 82 C5 81 63 82 C5 82 E0 ...
04830009 91 53 90 67 82 CC 97 CD 82 F0 90 55 82 82 8D 69 ...
0483000A 82 C1 82 C4 81 63 23 00 03 11 00 00 00 0A 42 ...
0483000B 00 04 06 00 82 BB 82 CC 8F EA 8F 8A 82 F0 96 DA ...
0483000C 8E 77 82 B5 82 C4 82 A2 82 BD 81 63 23 00 03 11 ...
0483000D 00 00 00 0A 43 00 0A 44 00 0A 45 00 0A 46 00 ...
0483000E 23 00 03 97 00 00 00 24 05 5B 24 FF 53 04 00 ...
0483000F 00 5D 5C 1E 24 FF 01 00 00 23 01 21 10 04 11 ...
04830010 00 04 28 53 49 52 4F 24 FF 00 00 00 24 FF 00 ...
04830011 00 00 24 FF 80 02 00 00 24 FF 00 01 00 24 ...
04830012 00 00 FF 00 00 24 FF 00 00 24 FF E3 03 00 ...
04830013 00 24 FF 00 00 00 00 24 FF 00 00 24 FF 00 ...

```

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

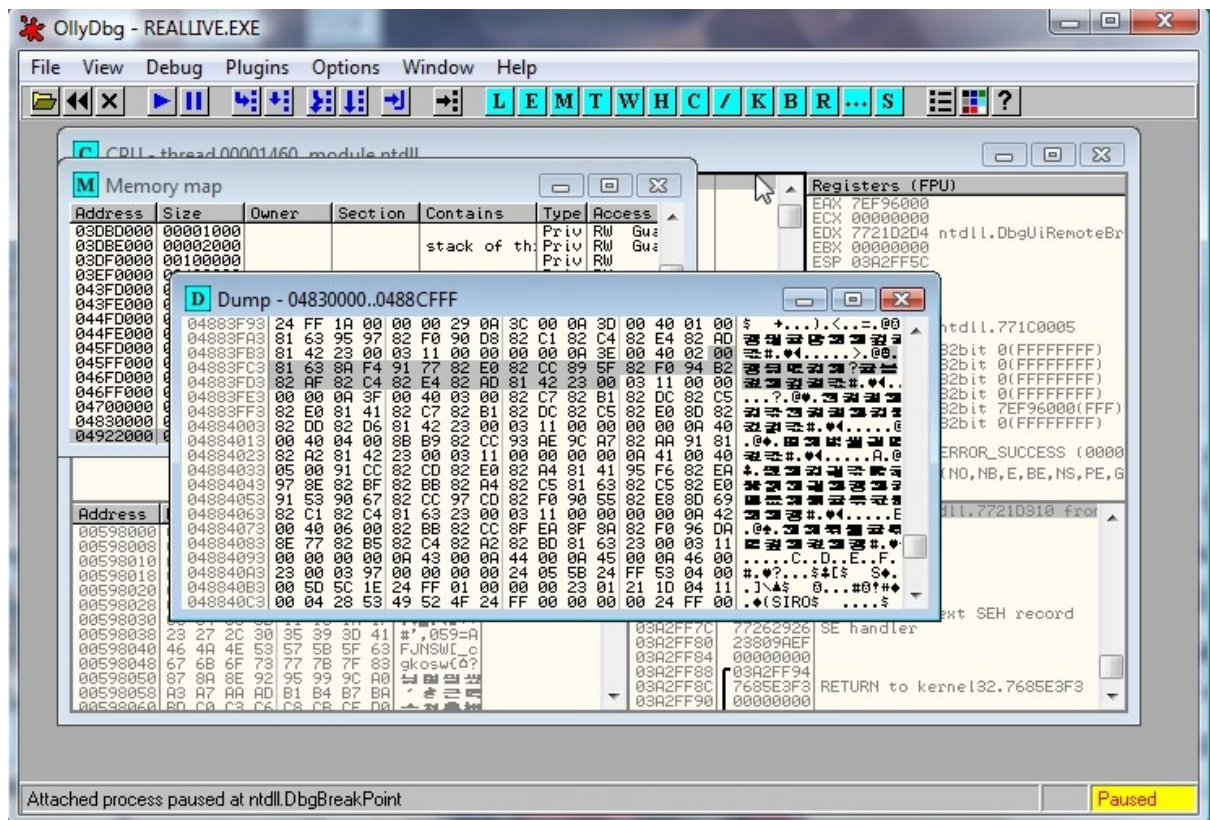
- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C

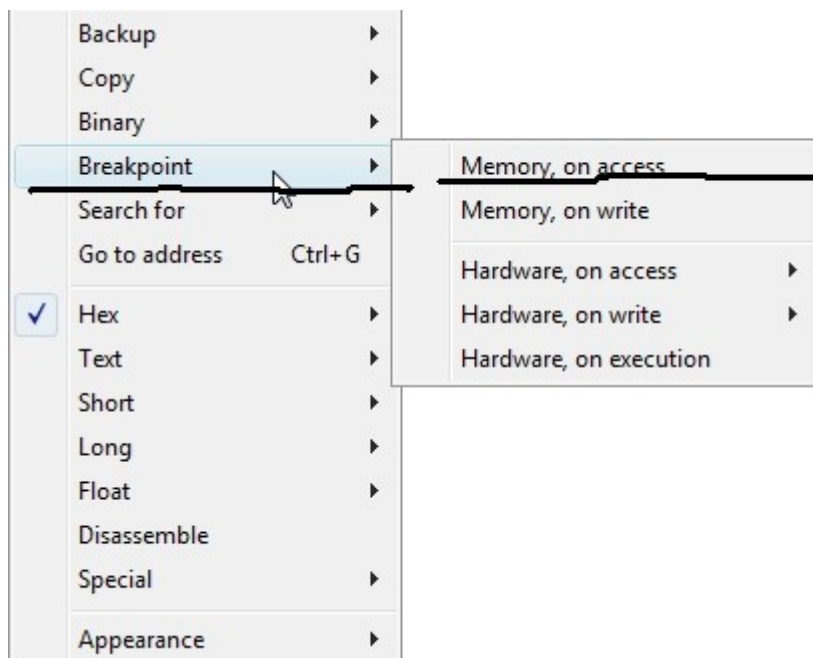
- Registers (FPU)** window showing the state of registers:

Register	Value
EAX	7EF96000
ECX	00000000
EDX	7721D2D4
EBX	00000000
ESP	03A2FF5C</

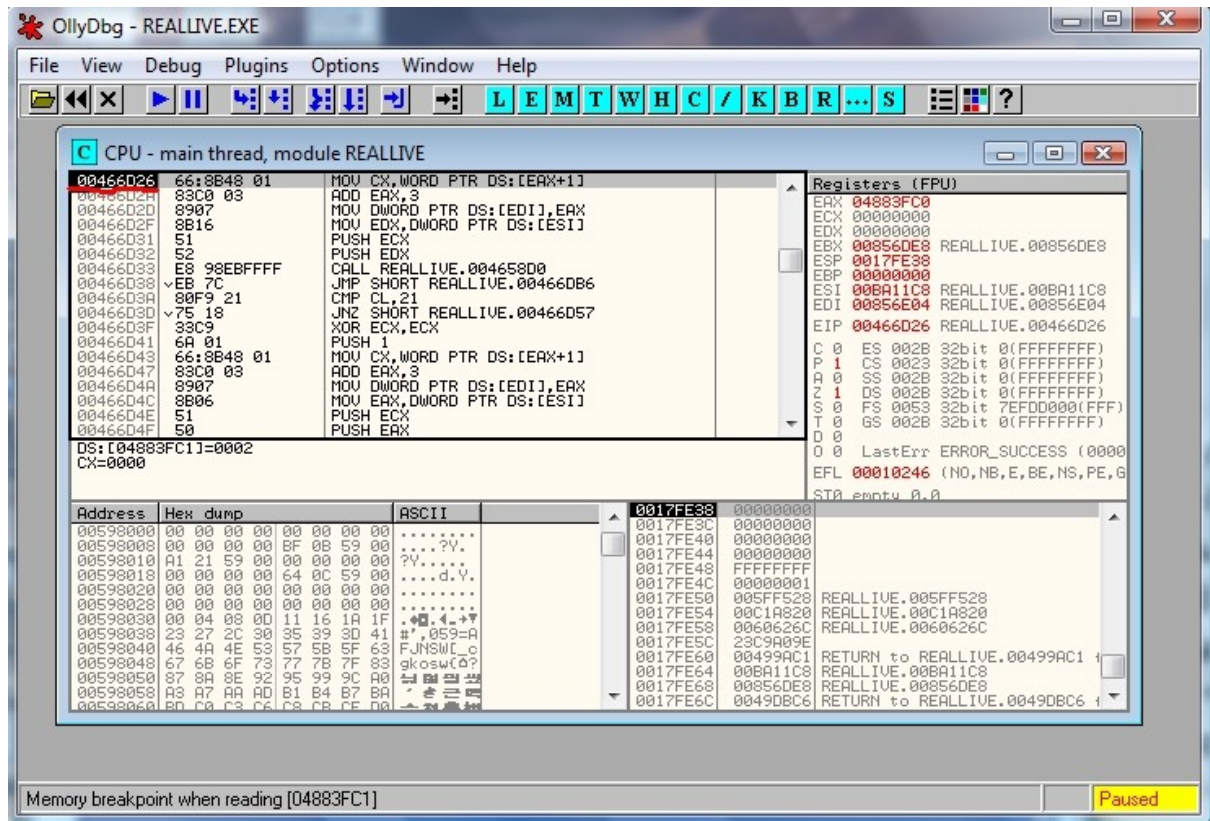
아자! 대사를 낚았습니다. 검은 줄 위로 81 63 8A...81 42 23 이 우리가 찾고 있는 대사.



대사 앞에 있는 00 부터 끝에 00 까지 선택 후, 마우스 오른쪽 버튼을 누릅니다.



브레이크 포인트를 이렇게 걸어줍니다.



저기 위에 빨간색으로 밑줄 쳐진게 대사 코드. 옆에 보면 MOV 어쩌구 저쩌구 밑으로 Add Eax 3. 더 밑으로 MOV 어쩌구 저쩌구.

즉, 최종코드는 466D26 EAX+3.

이제 선택지를 공략해봅시다.

선택지를 대사와 같이 울트라 에디트로 불러와 16 진수를 표기해 줍시다.

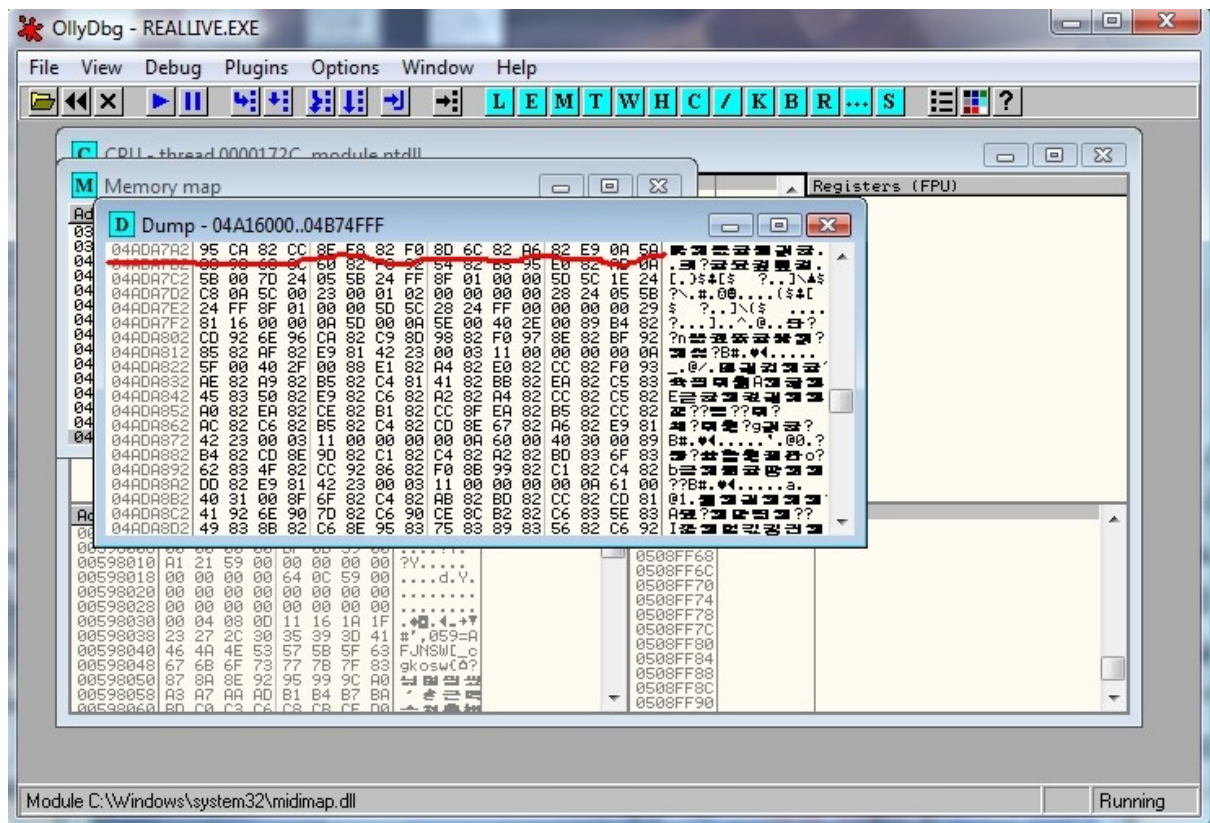
일단 게임을 끄시고 (울리를 종료해서) 다시 시작하시는 것을 추천합니다.

자, 이제 게임을 키시고, 울리를 먹이고, 게임을 선택지 전! 까지 진행해 줍시다.



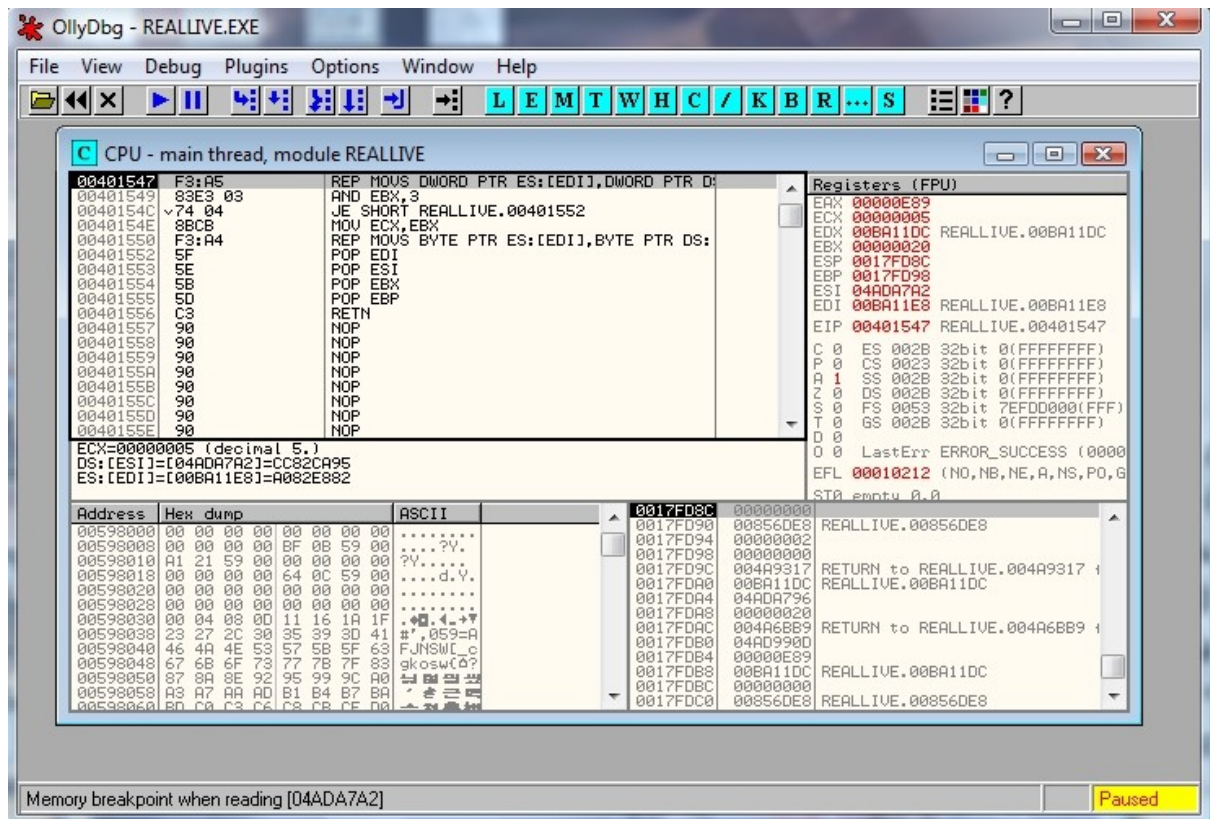
AIR 에선 이렇습니다.

대사와 같이 올리에서 선택지 문장을 낚습니다.

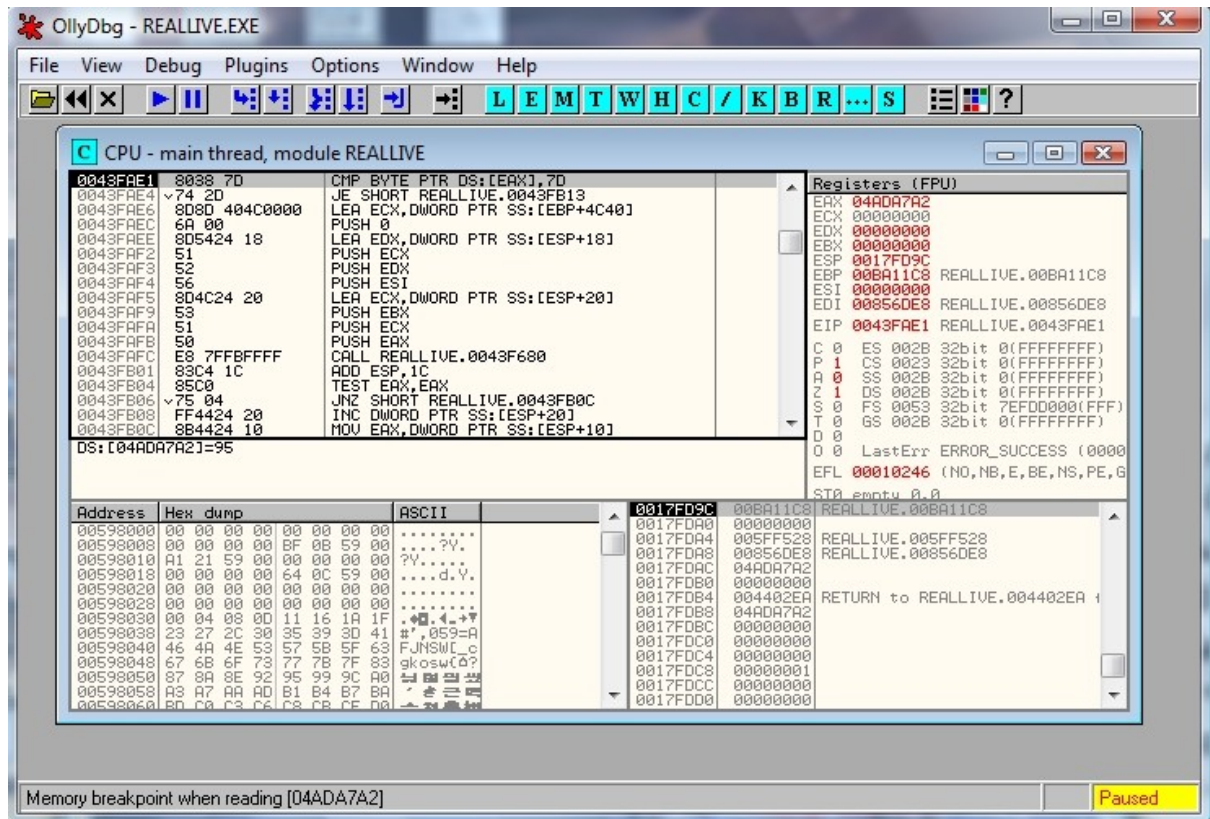


밑 줄쳐진게 선택지 문장. 대사는 00 으로 시작해 00 까지 선택해 주었지만, 선택지 문장은 00 아닌 숫자로 시작해 00 아닌 숫자로 끝납니다.

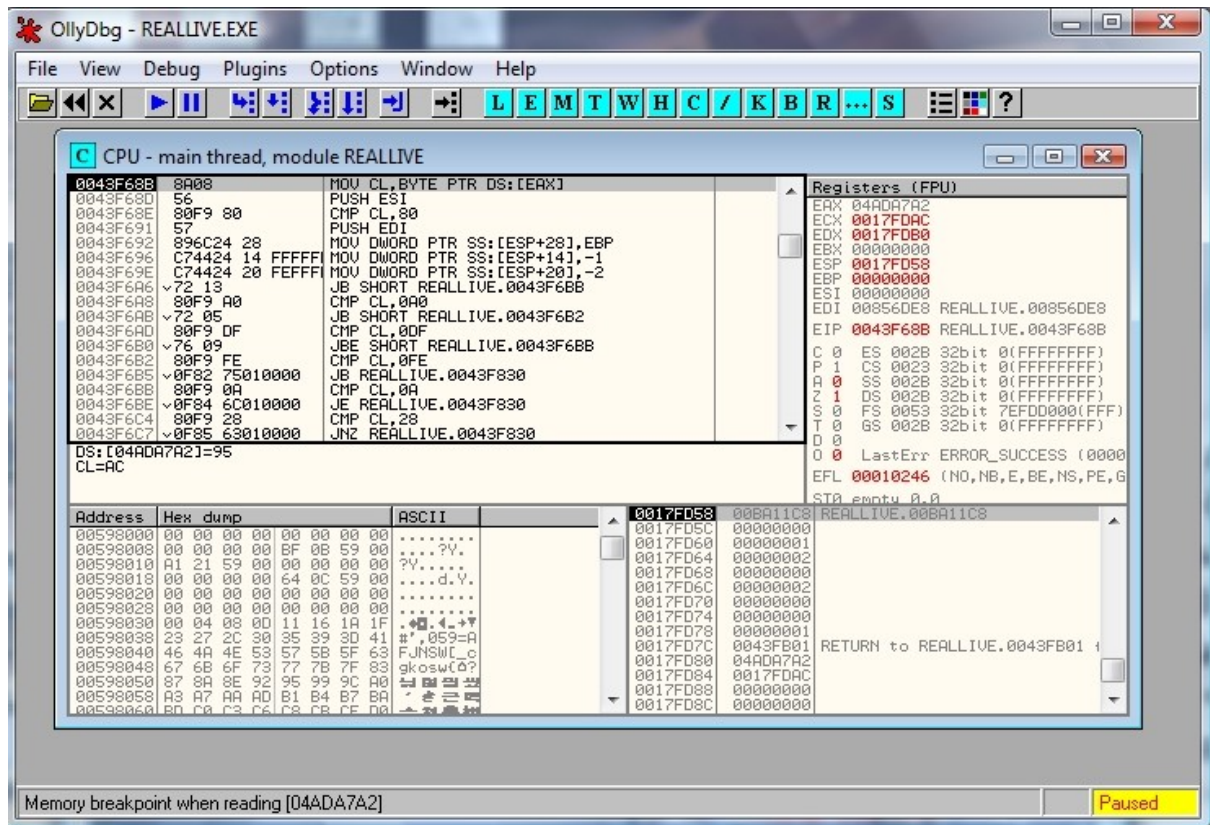
그 후, 브레이크 포인트를 걸면.....



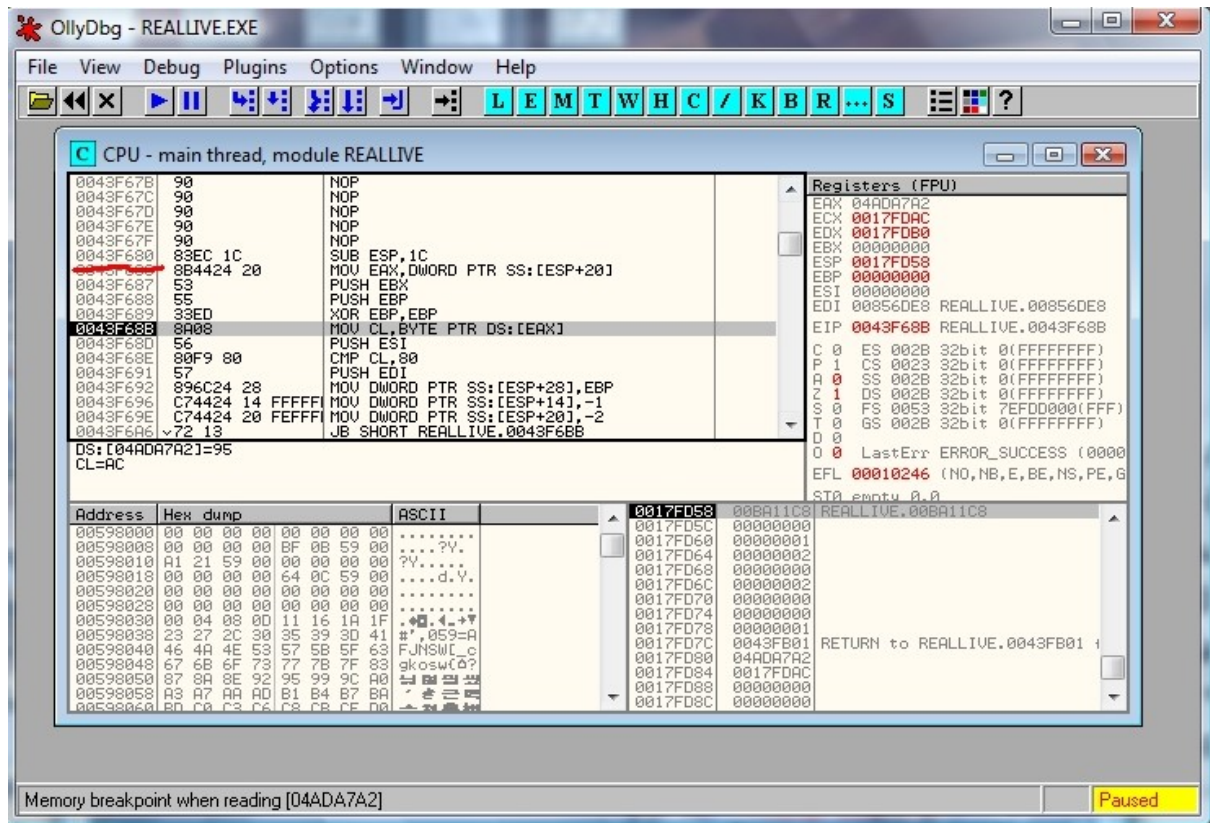
보면 REP 어쩌구 저쩌구 합니다. 요놈은 아니니...
F8을 누른후 F9을 누릅니다.



CMP 어쩌구 저쩌구..요놈도 아냐....
F8을 누른후 F9을 누릅니다.

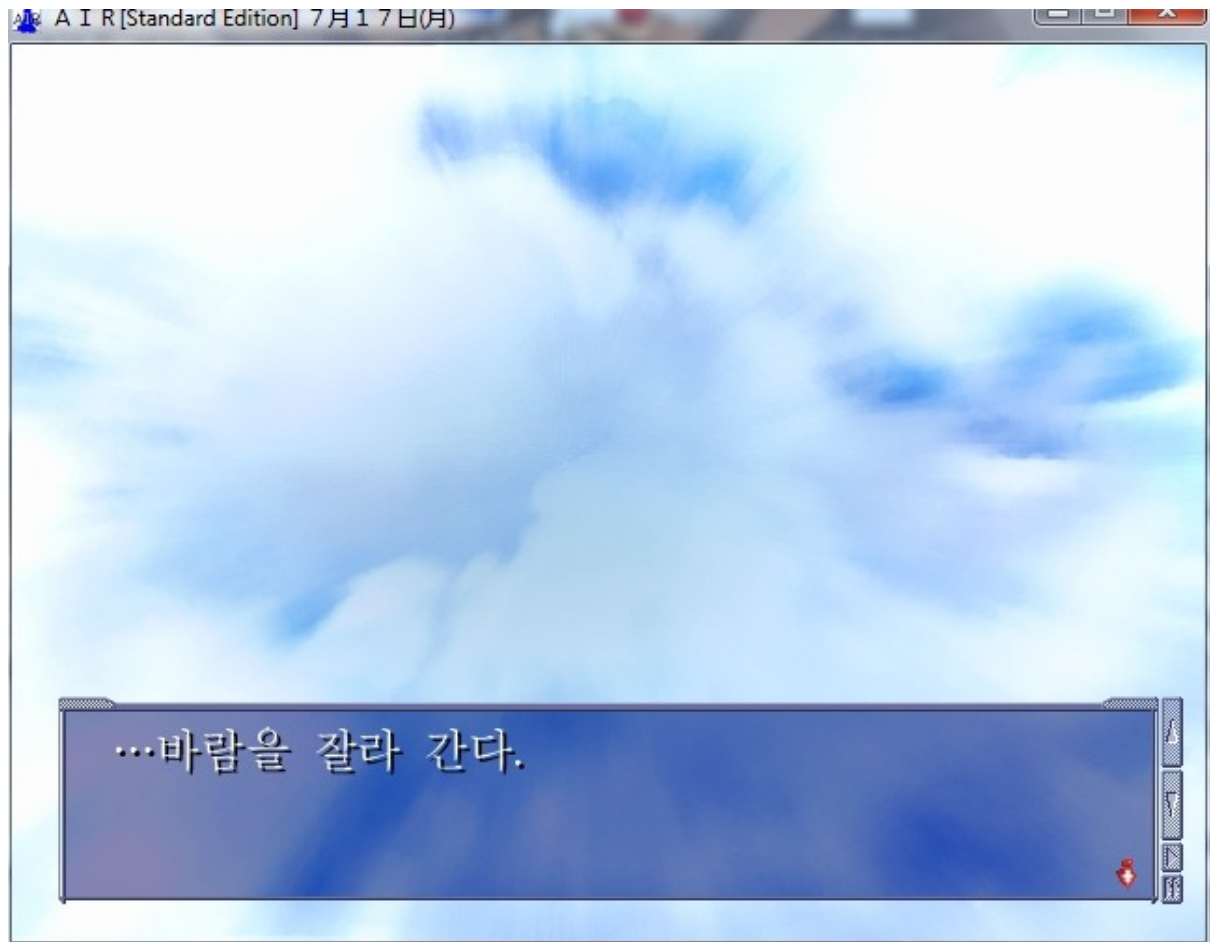


찾았다.
이제 스크를 업을 합니다. 조금만.
그럼...



밑 줄쳐진 곳이 코드. 43F680 ESP+4. ESP+4 는 옆에 SUB ESP, 1C 그리고 밑으로 어찌구 저찌구 하는 것을 계산해야 합니다. 전 못하므로 패스!!! 거의 대부분 ESP+4 을 쓰면 맞습니다.

이제 게임에 코드를 먹이면.....(스크립트 덮어쓰기와 RLcmd 필터, 한글폰트 불러오는 것 잊지말아주세요 ^^)



성공!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

끝입니다. 간단합니다.