Abex Crackme! #1

이지훈

목차

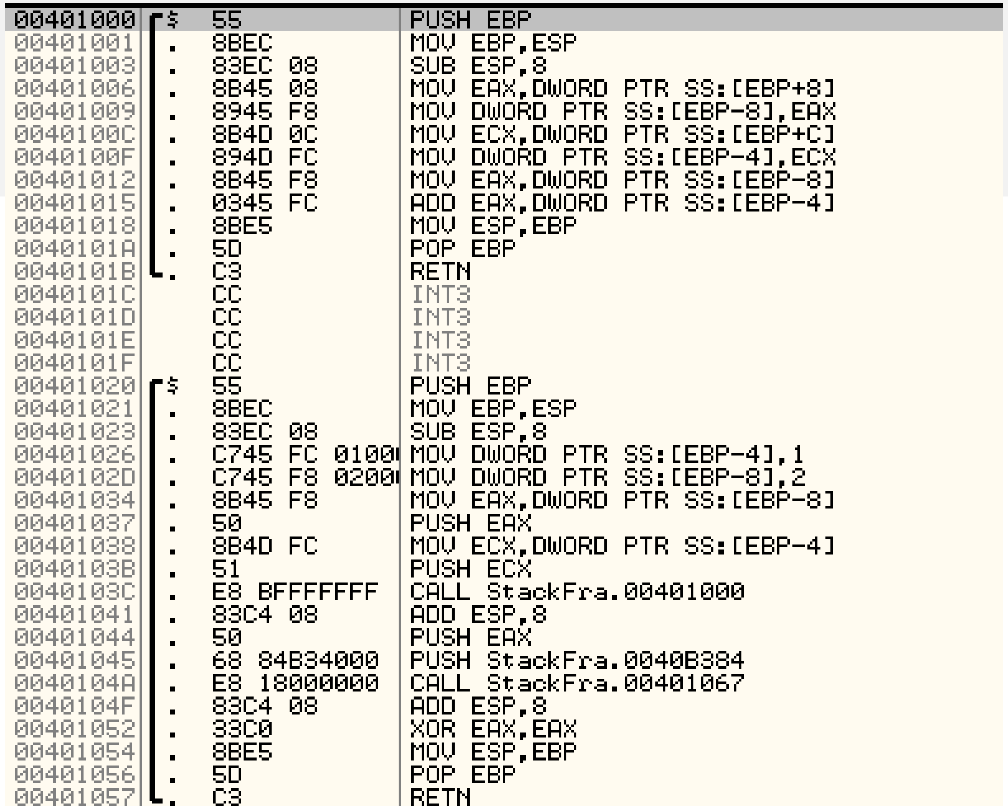
Abex Crackme! #1

Last Week

JE -> JMP crack

EAX, ESI crack

Last Week - Stack Frame.exe



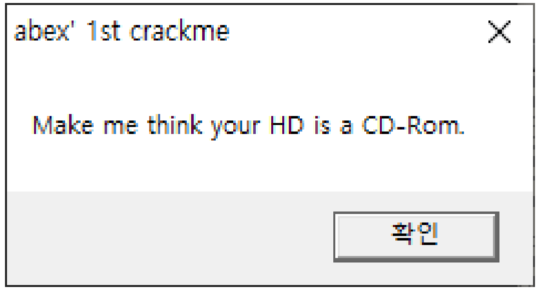
add()

```
00401000 ┌$  55            PUSH EBP
00401001 │.  8BEC          MOV EBP,ESP
00401003 │.  83EC 08       SUB ESP,8
00401006 │.  8B45 08       MOV EAX,DWORD PTR SS:[EBP+8]
00401009 │.  8945 F8       MOV DWORD PTR SS:[EBP-8],EAX
0040100C │.  8B4D 0C       MOV ECX,DWORD PTR SS:[EBP+C]
0040100F │.  894D FC       MOV DWORD PTR SS:[EBP-4],ECX
00401012 │.  8B45 F8       MOV EAX,DWORD PTR SS:[EBP-8]
00401015 │.  0345 FC       ADD EAX,DWORD PTR SS:[EBP-4]
00401018 │.  8BE5          MOV ESP,EBP
0040101A │.  5D            POP EBP
0040101B └.  C3            RETN
0040101C     CC            INT3
0040101D     CC            INT3
0040101E     CC            INT3
0040101F     CC            INT3
```

main()

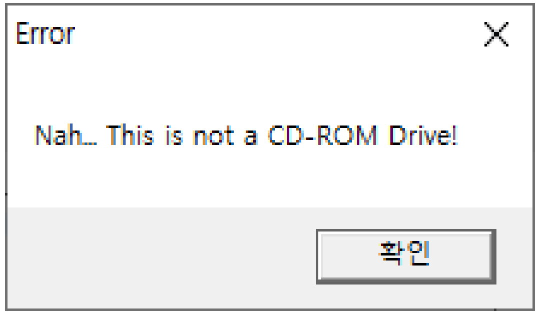
```
00401020 ┌$  55            PUSH EBP
00401021 │.  8BEC          MOV EBP,ESP
00401023 │.  83EC 08       SUB ESP,8
00401026 │.  C745 FC 0100  MOV DWORD PTR SS:[EBP-4],1
0040102D │.  C745 F8 0200  MOV DWORD PTR SS:[EBP-8],2
00401034 │.  8B45 F8       MOV EAX,DWORD PTR SS:[EBP-8]
00401037 │.  50            PUSH EAX
00401038 │.  8B4D FC       MOV ECX,DWORD PTR SS:[EBP-4]
0040103B │.  51            PUSH ECX
0040103C │.  E8 BFFFFFFF   CALL StackFra.00401000
00401041 │.  83C4 08       ADD ESP,8
00401044 │.  50            PUSH EAX
00401045 │.  68 84B34000   PUSH StackFra.0040B384
0040104A │.  E8 18000000   CALL StackFra.00401067
0040104F │.  83C4 08       ADD ESP,8
00401052 │.  33C0          XOR EAX,EAX
00401054 │.  8BE5          MOV ESP,EBP
00401056 │.  5D            POP EBP
00401057 └.  C3            RETN
```

abexcm1.exe



abex' 1st crackme

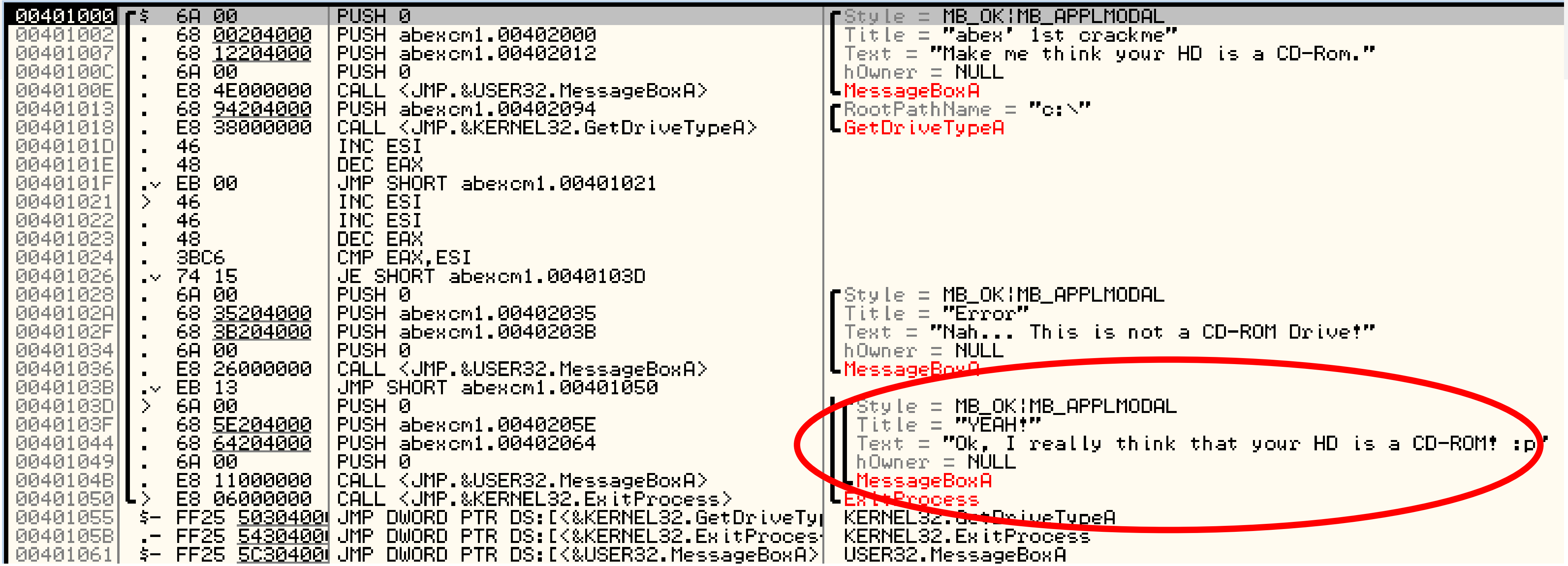
Make me think your HD is a CD-Rom.



Error

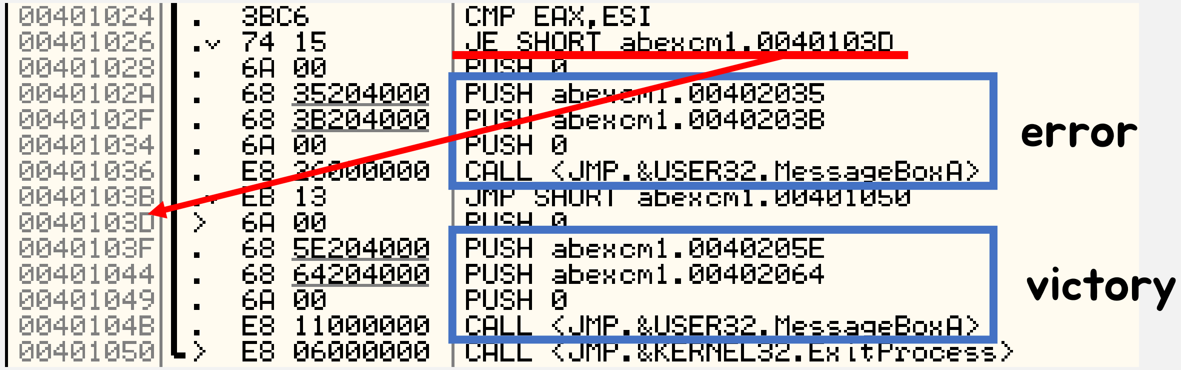
Nah... This is not a CD-ROM Drive!

abexcm1.exe



```
00401000  $   6A 00          PUSH 0                                     Style = MB_OK|MB_APPLMODAL
00401002  .   68 00204000    PUSH abexcm1.00402000                      Title = "abex' 1st crackme"
00401007  .   68 12204000    PUSH abexcm1.00402012                      Text = "Make me think your HD is a CD-Rom."
0040100C  .   6A 00          PUSH 0                                     hOwner = NULL
0040100E  .   E8 4E000000    CALL <JMP.&USER32.MessageBoxA>             MessageBoxA
00401013  .   68 94204000    PUSH abexcm1.00402094                      RootPathName = "c:\"
00401018  .   E8 38000000    CALL <JMP.&KERNEL32.GetDriveTypeA>         GetDriveTypeA
0040101D  .   46             INC ESI
0040101E  .   48             DEC EAX
0040101F  .v  EB 00          JMP SHORT abexcm1.00401021
00401021  >   46             INC ESI
00401022  .   46             INC ESI
00401023  .   48             DEC EAX
00401024  .   3BC6           CMP EAX,ESI
00401026  .v  74 15          JE SHORT abexcm1.0040103D
00401028  .   6A 00          PUSH 0                                     Style = MB_OK|MB_APPLMODAL
0040102A  .   68 35204000    PUSH abexcm1.00402035                      Title = "Error"
0040102F  .   68 3B204000    PUSH abexcm1.0040203B                      Text = "Nah... This is not a CD-ROM Drive!"
00401034  .   6A 00          PUSH 0                                     hOwner = NULL
00401036  .   E8 26000000    CALL <JMP.&USER32.MessageBoxA>             MessageBoxA
0040103B  .v  EB 13          JMP SHORT abexcm1.00401050
0040103D  >   6A 00          PUSH 0                                      Style = MB_OK|MB_APPLMODAL
0040103F  .   68 5E204000    PUSH abexcm1.0040205E                       Title = "YEAH!"
00401044  .   68 64204000    PUSH abexcm1.00402064                       Text = "Ok, I really think that your HD is a CD-ROM! :p"
00401049  .   6A 00          PUSH 0                                      hOwner = NULL
0040104B  .   E8 11000000    CALL <JMP.&USER32.MessageBoxA>              MessageBoxA
00401050  >   E8 06000000    CALL <JMP.&KERNEL32.ExitProcess>           ExitProcess
00401055  $-  FF25 5030400   JMP DWORD PTR DS:[<&KERNEL32.GetDriveTy    KERNEL32.GetDriveTypeA
0040105B  .-  FF25 5430400   JMP DWORD PTR DS:[<&KERNEL32.ExitProces    KERNEL32.ExitProcess
00401061  $-  FF25 5C30400   JMP DWORD PTR DS:[<&USER32.MessageBoxA>    USER32.MessageBoxA
```

abexcm1.exe

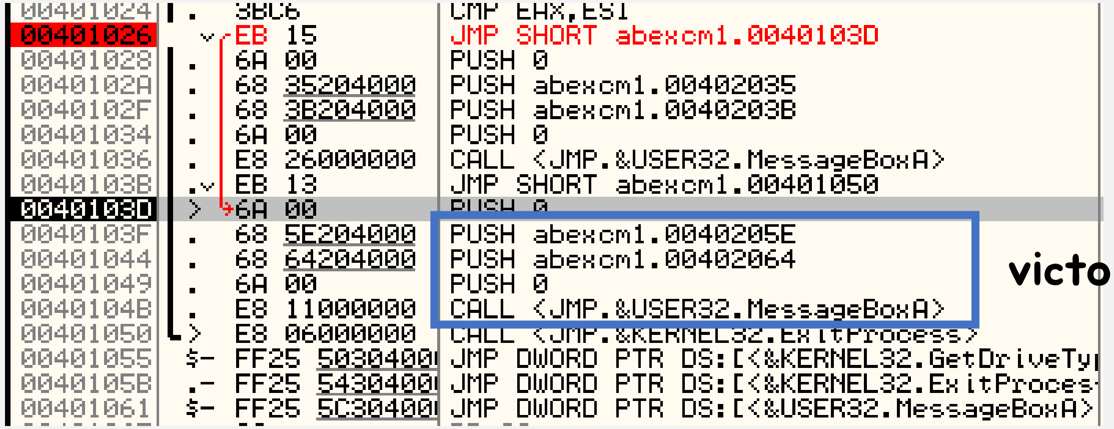


```
00401024  .   3BC6          CMP EAX,ESI
00401026  .v  74 15         JE SHORT abexcm1.0040103D
00401028  .   6A 00         PUSH 0
0040102A  .   68 35204000   PUSH abexcm1.00402035
0040102F  .   68 3B204000   PUSH abexcm1.0040203B
00401034  .   6A 00         PUSH 0
00401036  .   E8 26000000   CALL <JMP.&USER32.MessageBoxA>
0040103B  .   EB 13         JMP SHORT abexcm1.00401050
0040103D  >   6A 00         PUSH 0
0040103F  .   68 5E204000   PUSH abexcm1.0040205E
00401044  .   68 64204000   PUSH abexcm1.00402064
00401049  .   6A 00         PUSH 0
0040104B  .   E8 11000000   CALL <JMP.&USER32.MessageBoxA>
00401050  >   E8 06000000   CALL <JMP.&KERNEL32.ExitProcess>
```

CMP : compare, 주어진 두 개의 값을 비교한다

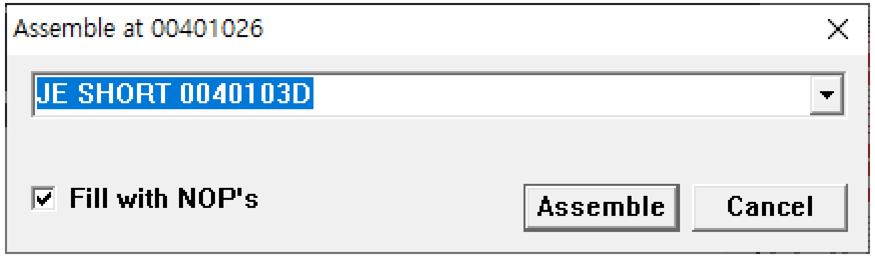
JE : Jump if Equal, CMP에서 두 개의 값이 같을 경우 지정한 주소로 JMP 한다
-> 두 개의 값이 다를 경우 밑의 있는 주소를 계속 진행한다

JE -> JMP



```
00401024  .   3BC6          CMP EAX,ESI
00401026    v EB 15         JMP SHORT abexcm1.0040103D
00401028  .   6A 00         PUSH 0
0040102A  .   68 35204000   PUSH abexcm1.00402035
0040102F  .   68 3B204000   PUSH abexcm1.0040203B
00401034  .   6A 00         PUSH 0
00401036  .   E8 26000000   CALL <JMP.&USER32.MessageBoxA>
0040103B  .v  EB 13         JMP SHORT abexcm1.00401050
0040103D  >   6A 00         PUSH 0
0040103F  .   68 5E204000   PUSH abexcm1.0040205E
00401044  .   68 64204000   PUSH abexcm1.00402064
00401049  .   6A 00         PUSH 0
0040104B  .   E8 11000000   CALL <JMP.&USER32.MessageBoxA>
00401050  >   E8 06000000   CALL <JMP.&KERNEL32.ExitProcess>
00401055  $-  FF25 5030400  JMP DWORD PTR DS:[<&KERNEL32.GetDriveTy
0040105B  .-  FF25 5430400  JMP DWORD PTR DS:[<&KERNEL32.ExitProces
00401061  $-  FF25 5C30400  JMP DWORD PTR DS:[<&USER32.MessageBoxA>
```

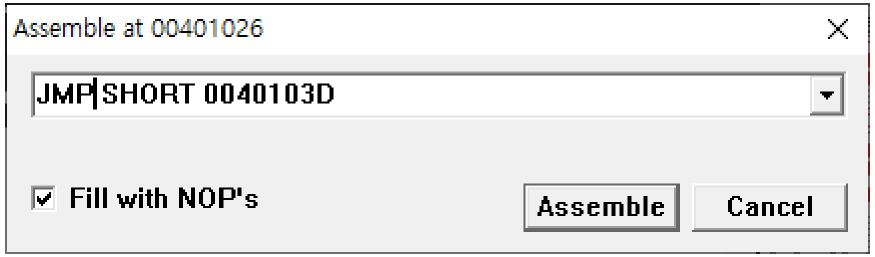
victory



Assemble at 00401026

JE SHORT 0040103D

☑ Fill with NOP's

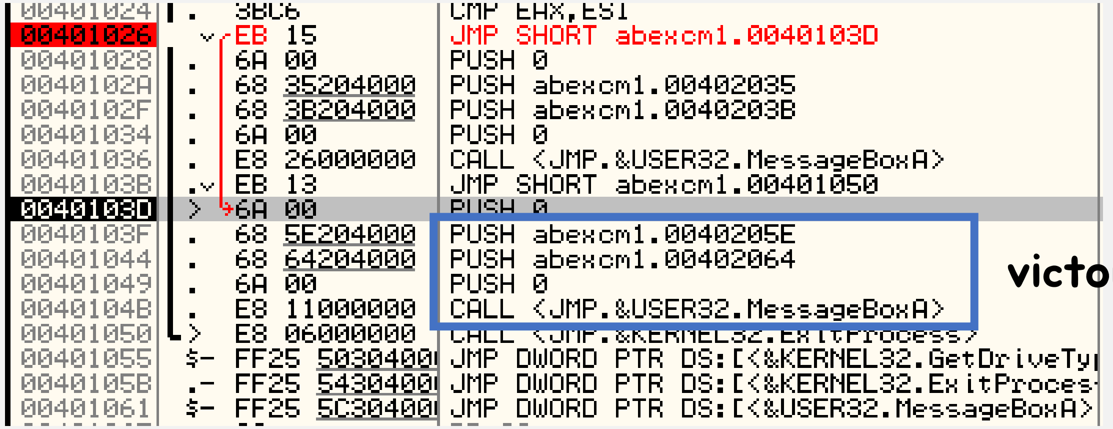


Assemble at 00401026

JMP SHORT 0040103D

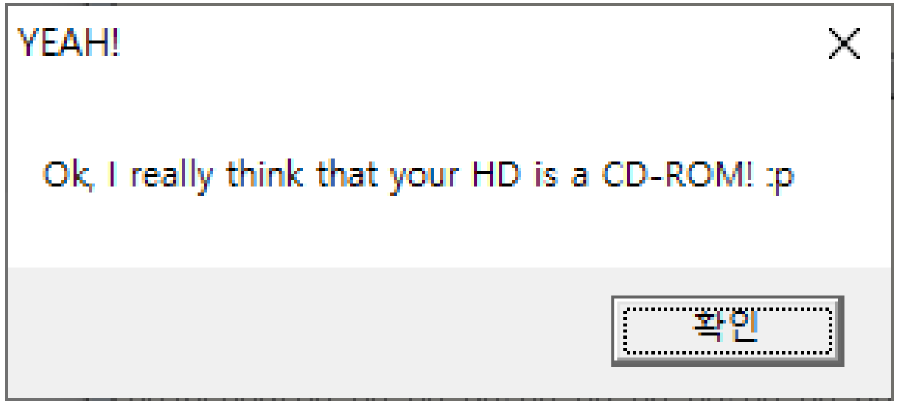
☑ Fill with NOP's

JE -> JMP



```
00401024  .    3BC6          CMP EAX,ESI
00401026    v  EB 15         JMP SHORT abexcm1.0040103D
00401028  .    6A 00         PUSH 0
0040102A  .    68 35204000   PUSH abexcm1.00402035
0040102F  .    68 3B204000   PUSH abexcm1.0040203B
00401034  .    6A 00         PUSH 0
00401036  .    E8 26000000   CALL <JMP.&USER32.MessageBoxA>
0040103B  .v   EB 13         JMP SHORT abexcm1.00401050
0040103D  >    6A 00         PUSH 0
0040103F  .    68 5E204000   PUSH abexcm1.0040205E
00401044  .    68 64204000   PUSH abexcm1.00402064
00401049  .    6A 00         PUSH 0
0040104B  .    E8 11000000   CALL <JMP.&USER32.MessageBoxA>
00401050  >    E8 06000000   CALL <JMP.&KERNEL32.ExitProcess>
00401055  $-   FF25 5030400  JMP DWORD PTR DS:[<&KERNEL32.GetDriveTy
0040105B  .-   FF25 5430400  JMP DWORD PTR DS:[<&KERNEL32.ExitProces
00401061  $-   FF25 5C30400  JMP DWORD PTR DS:[<&USER32.MessageBoxA>
```

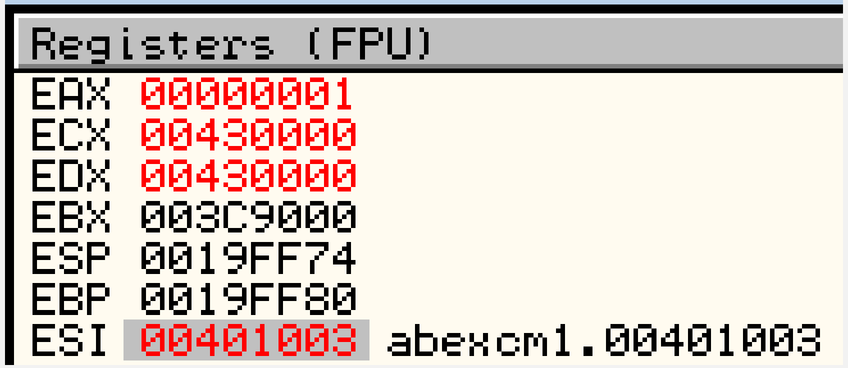
victory



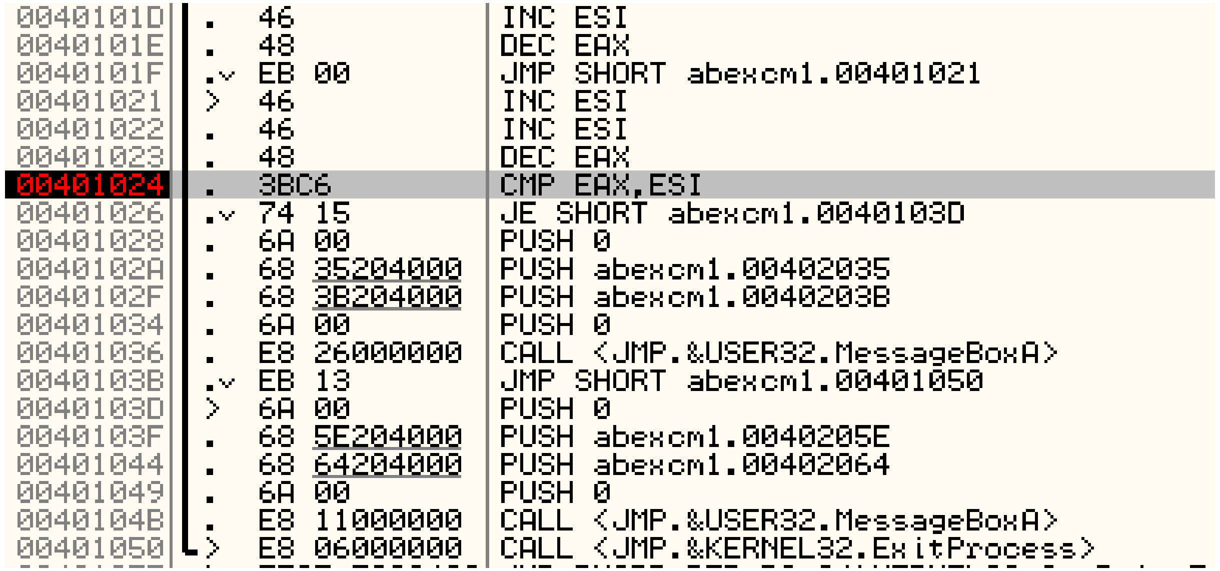
YEAH!

Ok, I really think that your HD is a CD-ROM! :p

abexcm1.exe

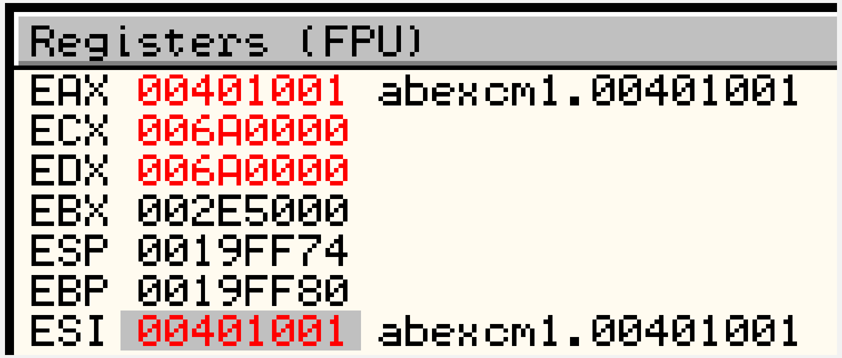


```
Registers (FPU)
EAX 00000001
ECX 00430000
EDX 00430000
EBX 003C9000
ESP 0019FF74
EBP 0019FF80
ESI 00401003 abexcm1.00401003
```

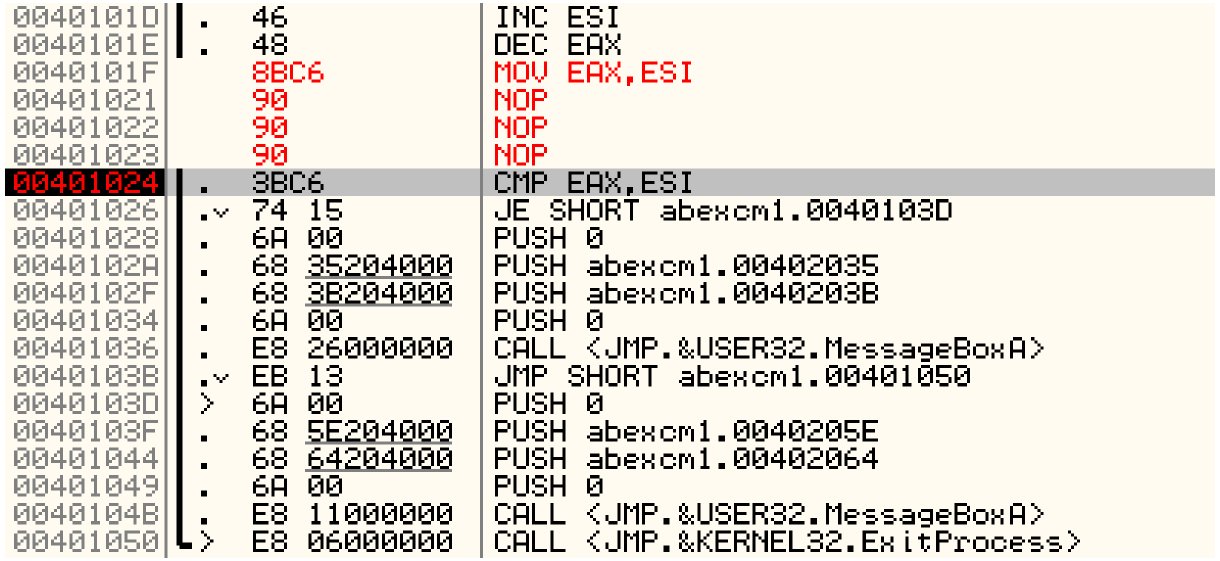


```
0040101D  .   46          INC ESI
0040101E  .   48          DEC EAX
0040101F  .v  EB 00       JMP SHORT abexcm1.00401021
00401021  >   46          INC ESI
00401022  .   46          INC ESI
00401023  .   48          DEC EAX
00401024  .   3BC6        CMP EAX,ESI
00401026  .v  74 15       JE SHORT abexcm1.0040103D
00401028  .   6A 00       PUSH 0
0040102A  .   68 35204000 PUSH abexcm1.00402035
0040102F  .   68 3B204000 PUSH abexcm1.0040203B
00401034  .   6A 00       PUSH 0
00401036  .   E8 26000000 CALL <JMP.&USER32.MessageBoxA>
0040103B  .v  EB 13       JMP SHORT abexcm1.00401050
0040103D  >   6A 00       PUSH 0
0040103F  .   68 5E204000 PUSH abexcm1.0040205E
00401044  .   68 64204000 PUSH abexcm1.00402064
00401049  .   6A 00       PUSH 0
0040104B  .   E8 11000000 CALL <JMP.&USER32.MessageBoxA>
00401050  >   E8 06000000 CALL <JMP.&KERNEL32.ExitProcess>
```

EAX, ESI

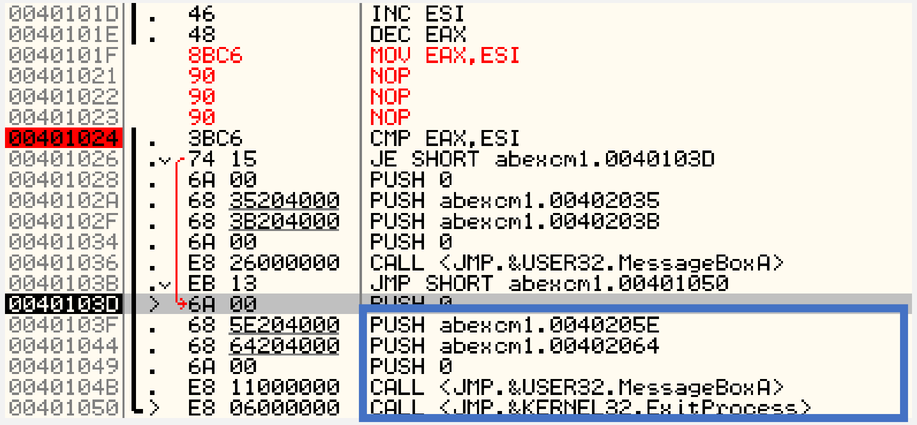


```
Registers (FPU)
EAX 00401001 abexcm1.00401001
ECX 006A0000
EDX 006A0000
EBX 002E5000
ESP 0019FF74
EBP 0019FF80
ESI 00401001 abexcm1.00401001
```



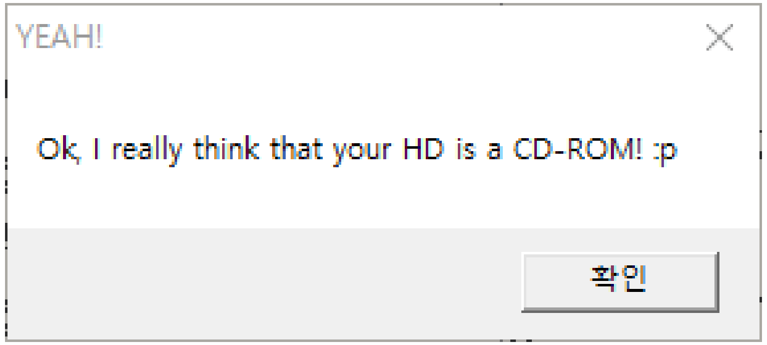
```
0040101D  .  46           INC ESI
0040101E  .  48           DEC EAX
0040101F     8BC6         MOV EAX,ESI
00401021     90           NOP
00401022     90           NOP
00401023     90           NOP
00401024  .  3BC6         CMP EAX,ESI
00401026  .v 74 15        JE SHORT abexcm1.0040103D
00401028  .  6A 00        PUSH 0
0040102A  .  68 35204000  PUSH abexcm1.00402035
0040102F  .  68 3B204000  PUSH abexcm1.0040203B
00401034  .  6A 00        PUSH 0
00401036  .  E8 26000000  CALL <JMP.&USER32.MessageBoxA>
0040103B  .v EB 13        JMP SHORT abexcm1.00401050
0040103D  >  6A 00        PUSH 0
0040103F  .  68 5E204000  PUSH abexcm1.0040205E
00401044  .  68 64204000  PUSH abexcm1.00402064
00401049  .  6A 00        PUSH 0
0040104B  .  E8 11000000  CALL <JMP.&USER32.MessageBoxA>
00401050  >  E8 06000000  CALL <JMP.&KERNEL32.ExitProcess>
```

EAX, ESI



```
0040101D  .   46            INC ESI
0040101E  .   48            DEC EAX
0040101F      8BC6          MOV EAX,ESI
00401021      90            NOP
00401022      90            NOP
00401023      90            NOP
00401024  .   3BC6          CMP EAX,ESI
00401026  .v  74 15         JE SHORT abexcm1.0040103D
00401028  .   6A 00         PUSH 0
0040102A  .   68 35204000   PUSH abexcm1.00402035
0040102F  .   68 3B204000   PUSH abexcm1.0040203B
00401034  .   6A 00         PUSH 0
00401036  .   E8 26000000   CALL <JMP.&USER32.MessageBoxA>
0040103B  .v  EB 13         JMP SHORT abexcm1.00401050
0040103D  >   6A 00         PUSH 0
0040103F  .   68 5E204000   PUSH abexcm1.0040205E
00401044  .   68 64204000   PUSH abexcm1.00402064
00401049  .   6A 00         PUSH 0
0040104B  .   E8 11000000   CALL <JMP.&USER32.MessageBoxA>
00401050  >   E8 06000000   CALL <JMP.&KERNEL32.ExitProcess>
```

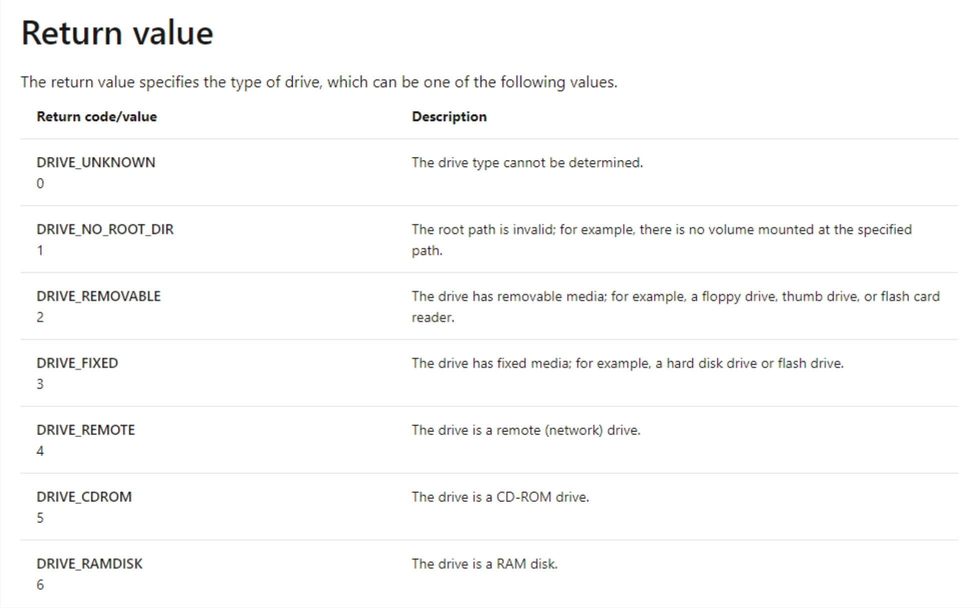
victory



YEAH!

Ok, I really think that your HD is a CD-ROM! :p

Etc..



Return value

The return value specifies the type of drive, which can be one of the following values.

Return code/value	Description
DRIVE_UNKNOWN 0	The drive type cannot be determined.
DRIVE_NO_ROOT_DIR 1	The root path is invalid; for example, there is no volume mounted at the specified path.
DRIVE_REMOVABLE 2	The drive has removable media; for example, a floppy drive, thumb drive, or flash card reader.
DRIVE_FIXED 3	The drive has fixed media; for example, a hard disk drive or flash drive.
DRIVE_REMOTE 4	The drive is a remote (network) drive.
DRIVE_CDROM 5	The drive is a CD-ROM drive.
DRIVE_RAMDISK 6	The drive is a RAM disk.

- GetDriveTypeA 명령어에서 CD-rom 을 인식시 리턴하는 EAX값 넣기

- INC, DEC 명령어를 수정하여 같은 값으로 만들기

. . . -> 최소한의 수정으로 프로그램 내부 루틴에 영향을 줄이는 방법을 모색!

Q & A