Họ và tên: Trần Trí Dũng

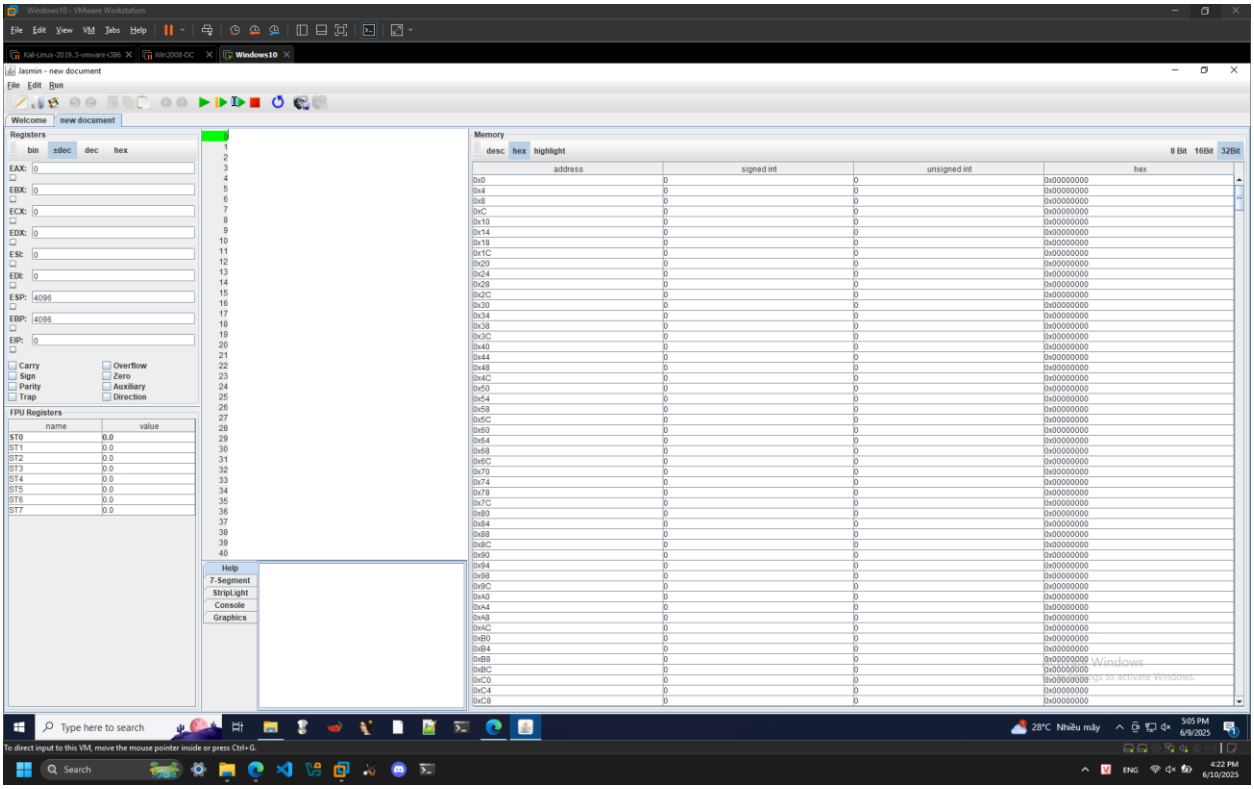
Mã số sinh viên: HE190680

Lớp: IA1901

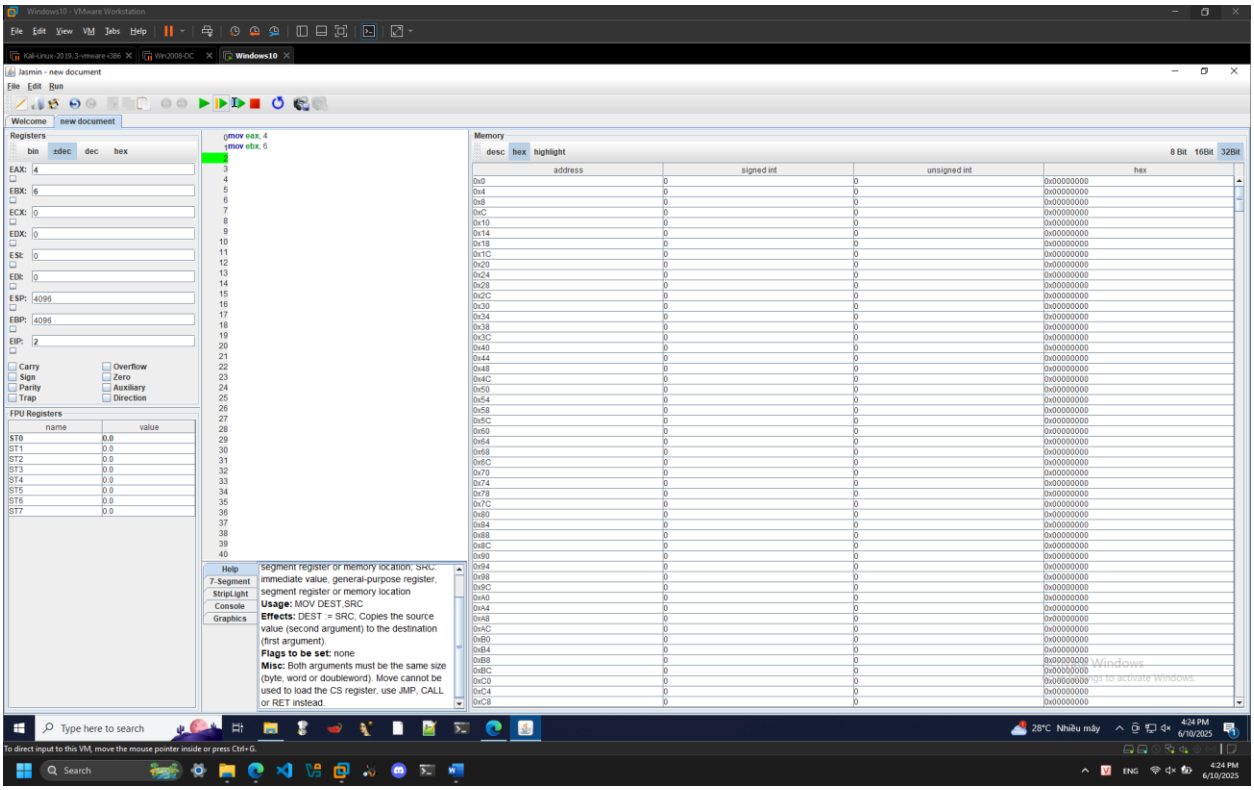
Lab 9

1. Using Jasmin to run x86 Assembly Code

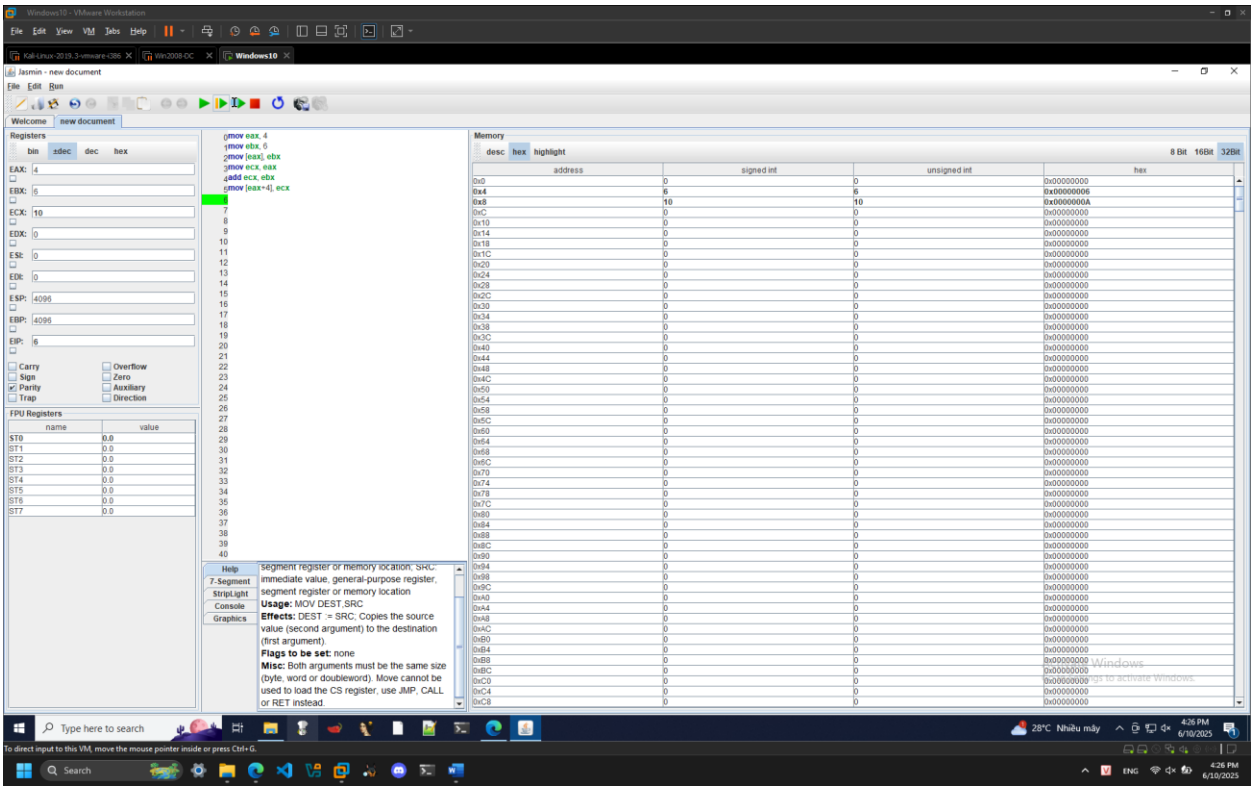
Understanding the Jasmin Window:



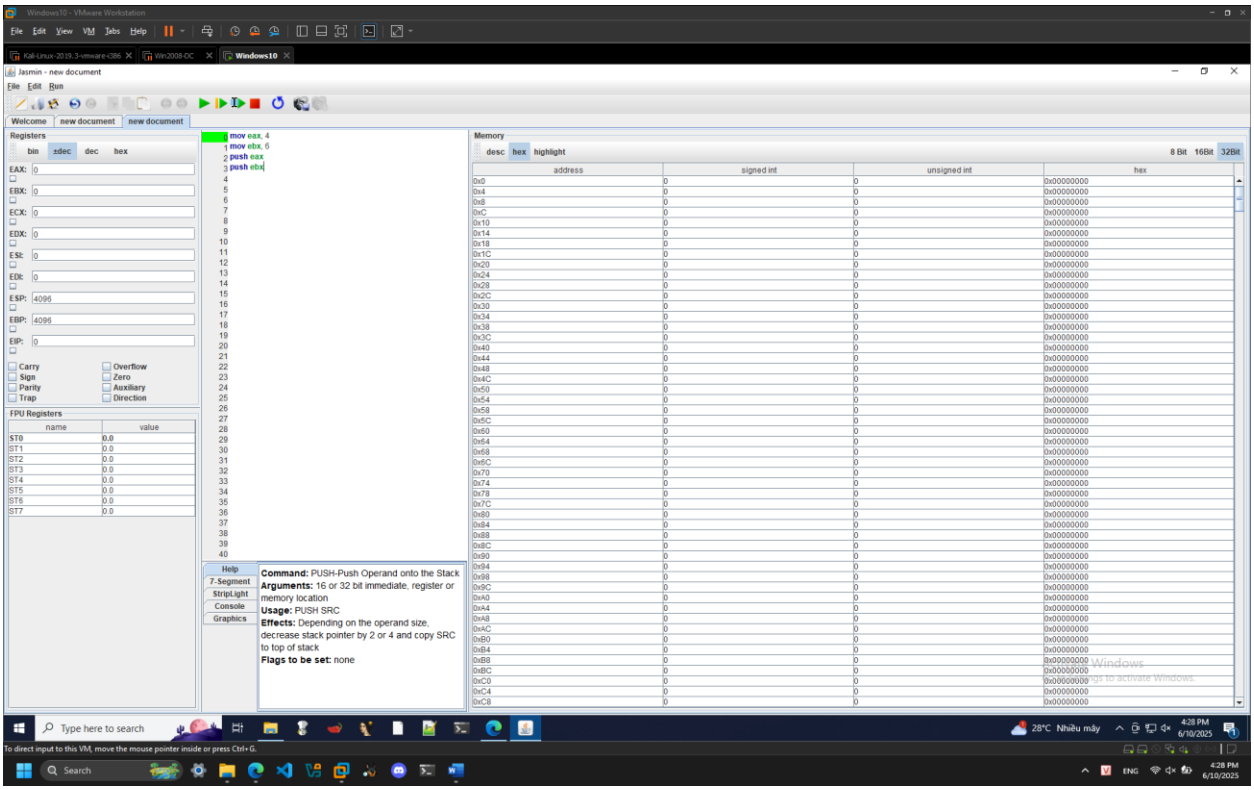
Using mov Instructions:



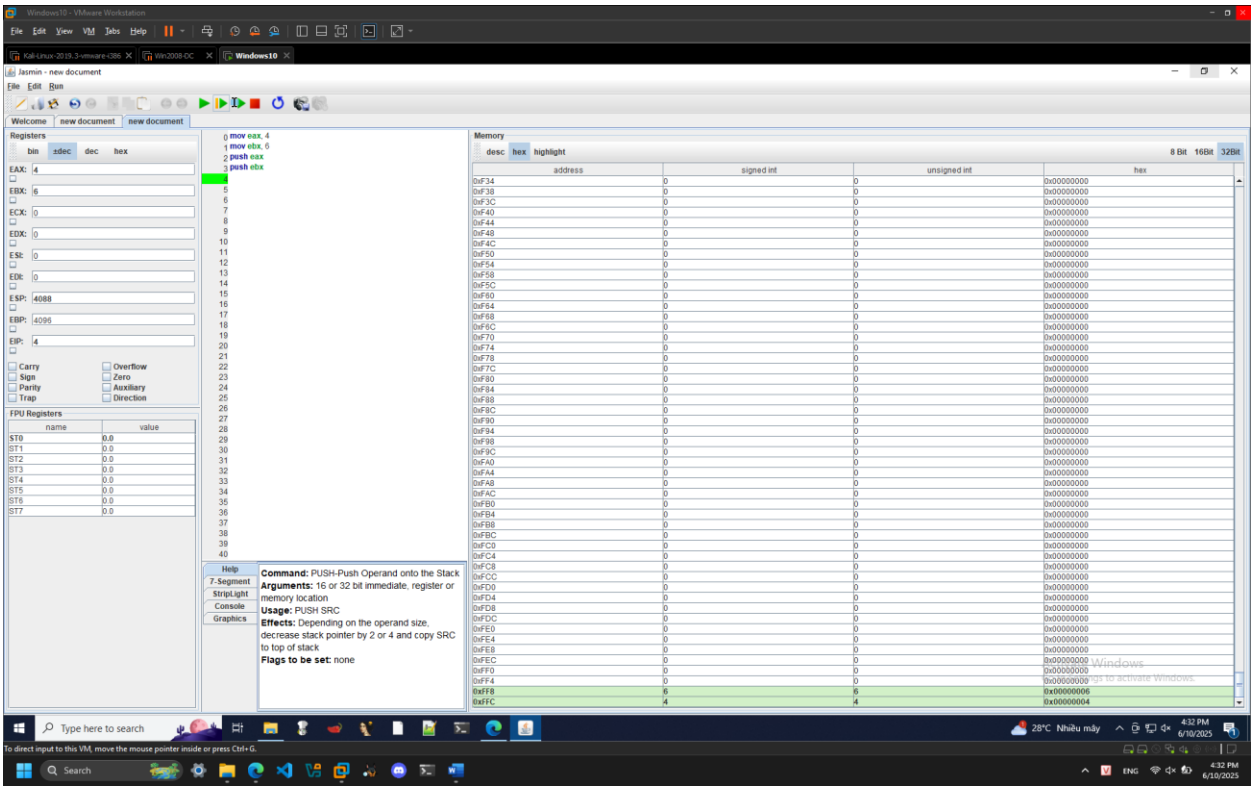
Storing Results in Memory:



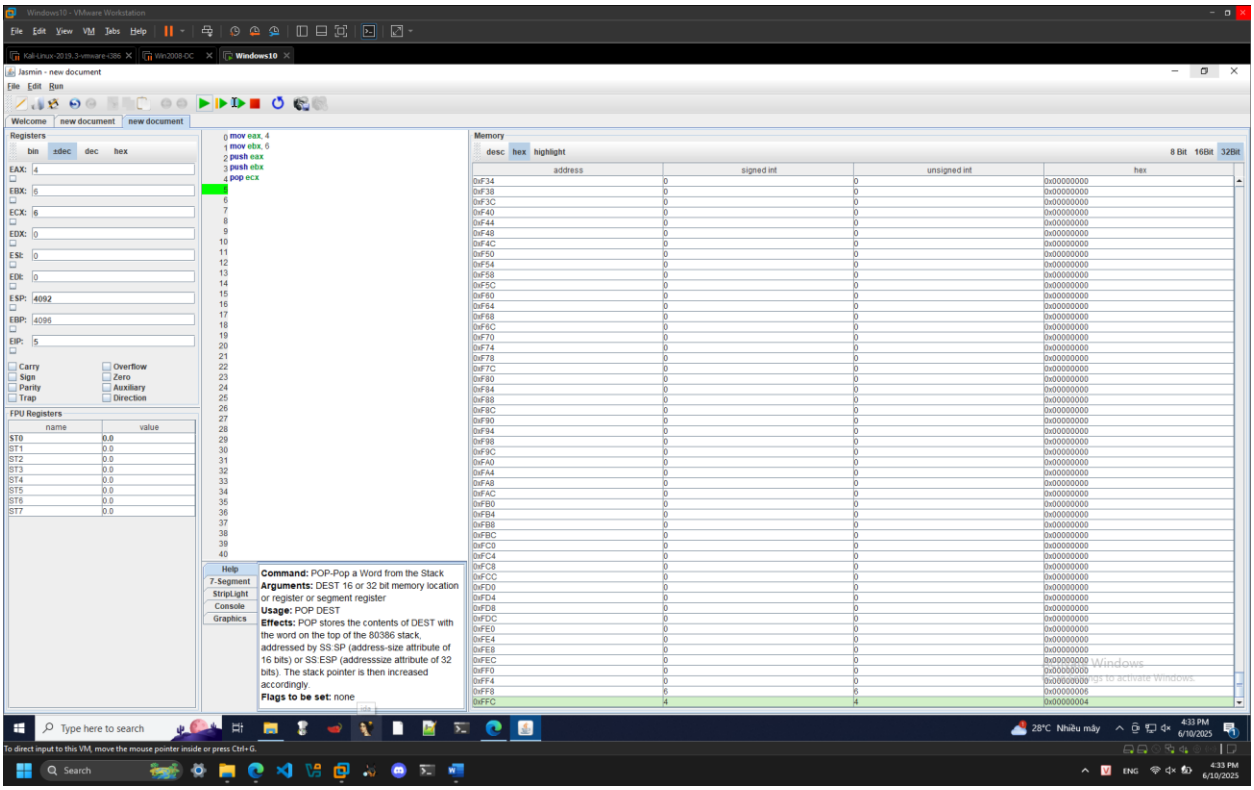
Using the Stack:



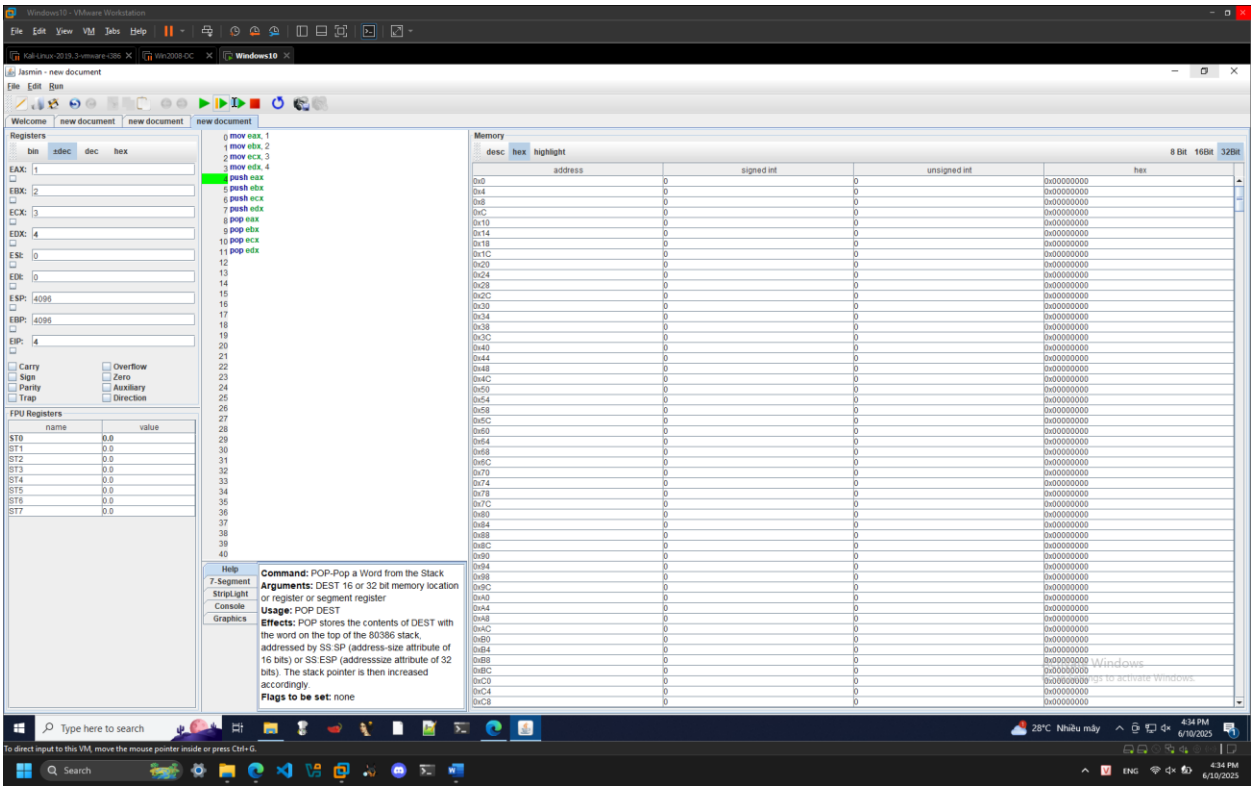
Understanding Push:

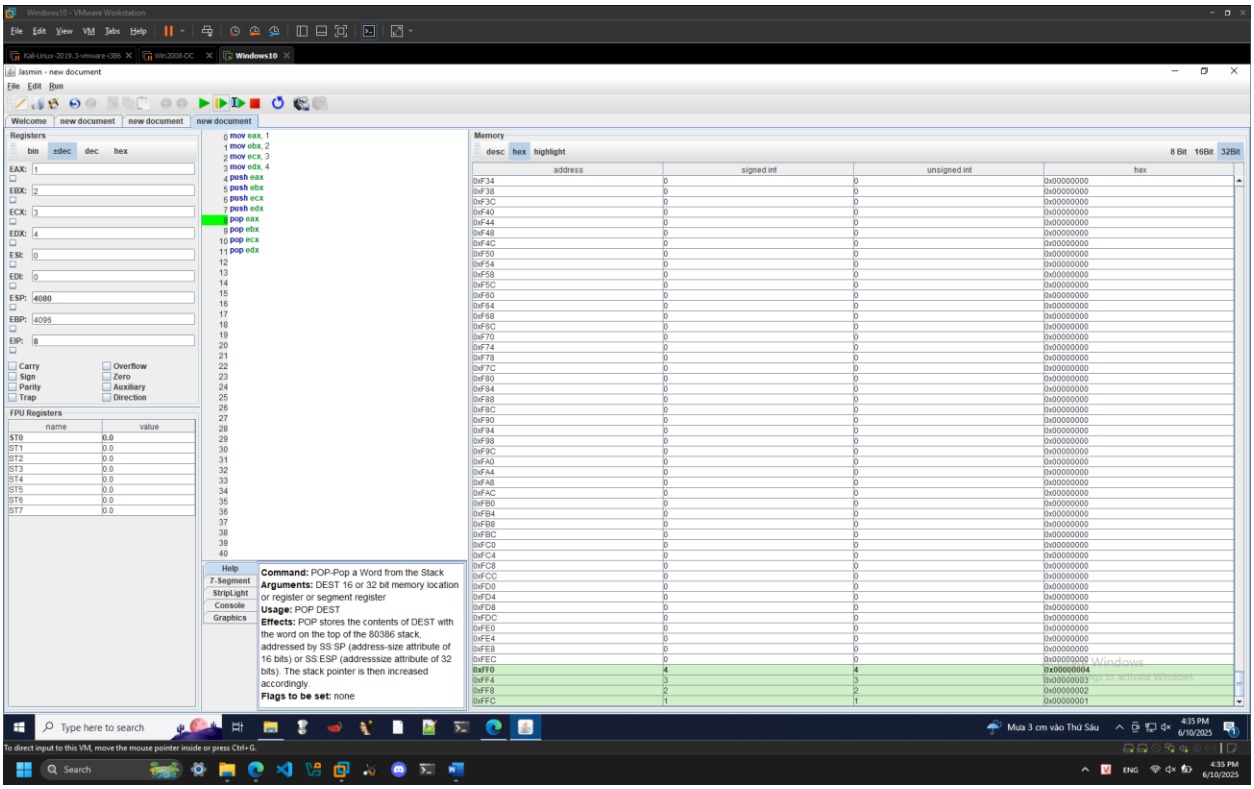


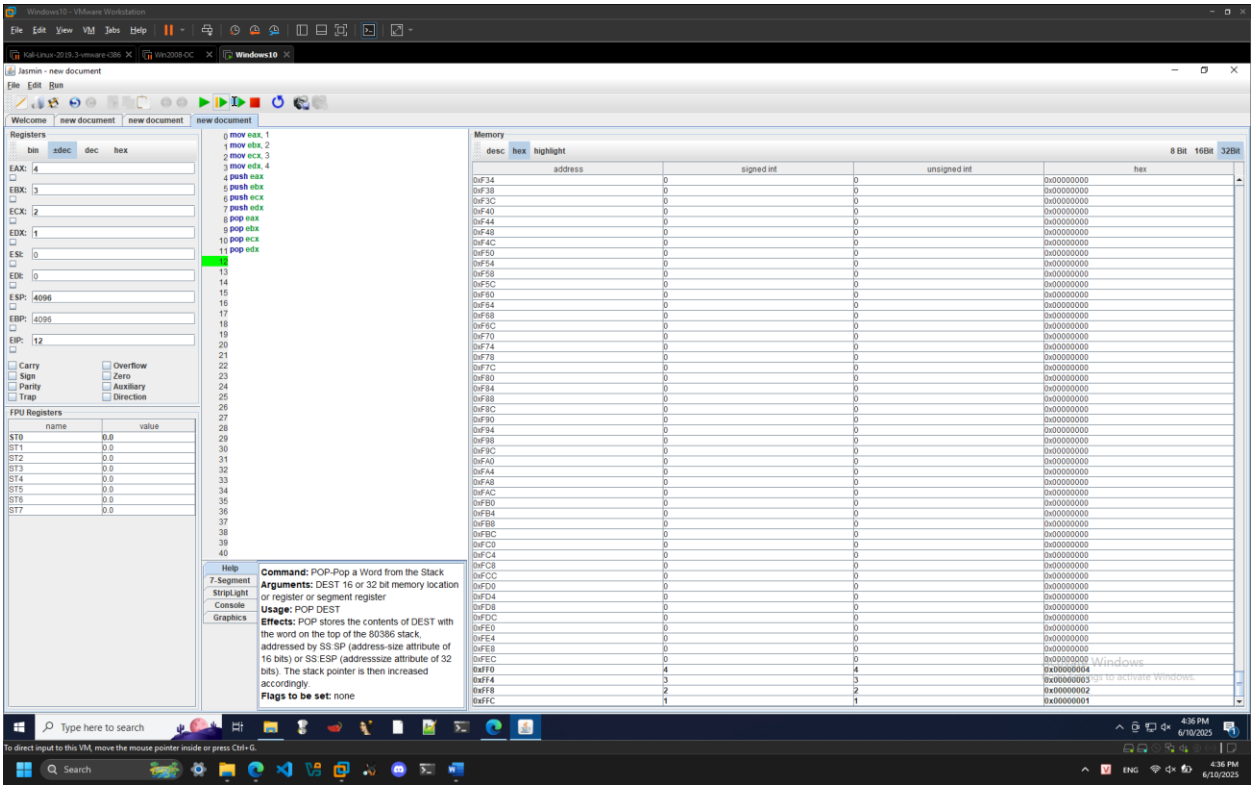
Understanding Pop:



Reversing a Sequence:

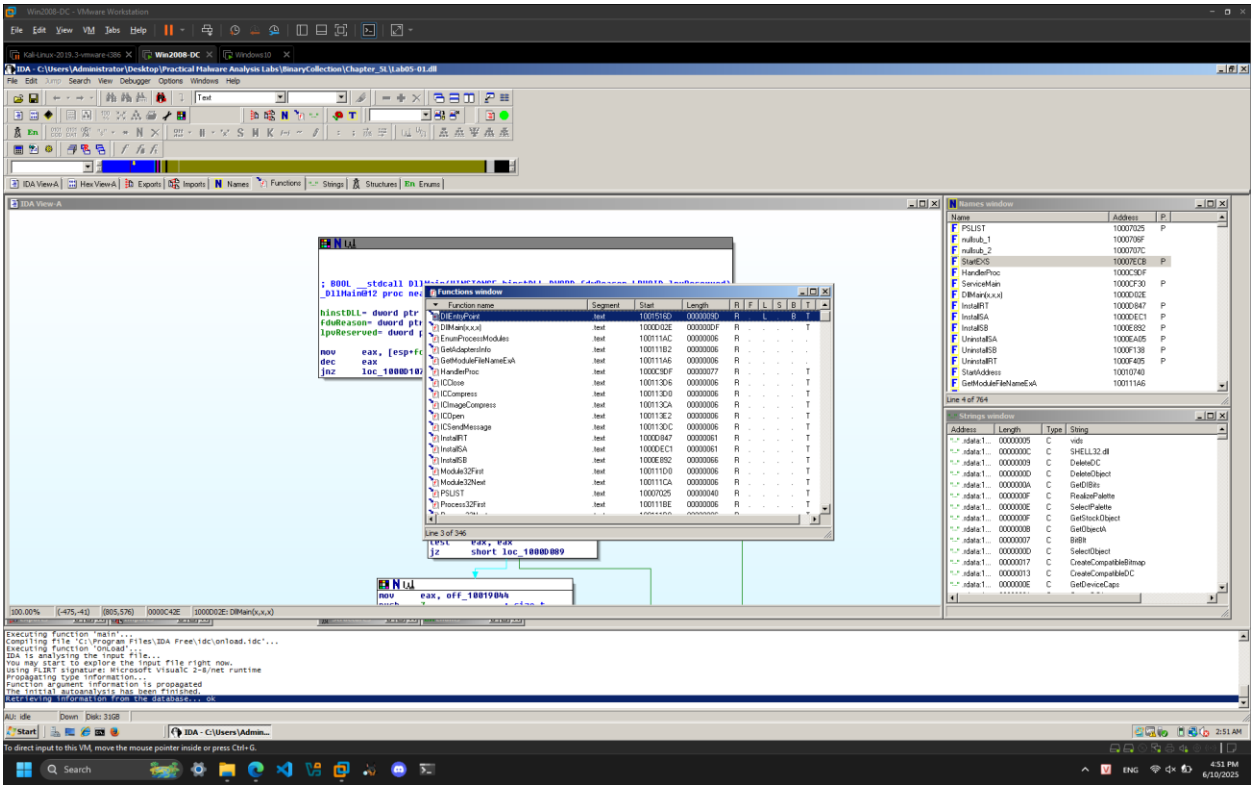



Jasmin - new document
File Edit Run
Welcome
new document
new document
new document
Registers
bin ±dec dec hex
EAX: 1
EBX: 2
ECX: 3
EDX: 4
ESI: 0
EDI: 0
ESP: 4080
EBP: 4096
EIP: 8
Carry Overflow
Sign Zero
Parity Auxiliary
Trap Direction
FPU Registers
name value
ST0 0.0
ST1 0.0
ST2 0.0
ST3 0.0
ST4 0.0
ST5 0.0
ST6 0.0
ST7 0.0
0 mov eax, 1
1 mov ebx, 2
2 mov ecx, 3
3 mov edx, 4
4 push eax
5 push ebx
6 push ecx
7 push edx
8 pop eax
9 pop ebx
10 pop ecx
11 pop edx
Help
7-Segment
StripLight
Console
Graphics
Command: POP-Pop a Word from the Stack
Arguments: DEST 16 or 32 bit memory location or register or segment register
Usage: POP DEST
Effects: POP stores the contents of DEST with the word on the top of the 80386 stack, addressed by SS:SP (address-size attribute of 16 bits) or SS:ESP (addresssize attribute of 32 bits). The stack pointer is then increased accordingly.
Flags to be set: none
Memory
desc hex highlight
8 Bit 16Bit 32Bit
address signed int unsigned int hex
0xF34 0 0 0x00000000
0xF38 0 0 0x00000000
0xF3C 0 0 0x00000000
0xF40 0 0 0x00000000
0xF44 0 0 0x00000000
0xF48 0 0 0x00000000
0xF4C 0 0 0x00000000
0xF50 0 0 0x00000000
0xF54 0 0 0x00000000
0xF58 0 0 0x00000000
0xF5C 0 0 0x00000000
0xF60 0 0 0x00000000
0xF64 0 0 0x00000000
0xF68 0 0 0x00000000
0xF6C 0 0 0x00000000
0xF70 0 0 0x00000000
0xF74 0 0 0x00000000
0xF78 0 0 0x00000000
0xF7C 0 0 0x00000000
0xF80 0 0 0x00000000
0xF84 0 0 0x00000000
0xF88 0 0 0x00000000
0xF8C 0 0 0x00000000
0xF90 0 0 0x00000000
0xF94 0 0 0x00000000
0xF98 0 0 0x00000000
0xF9C 0 0 0x00000000
0xFA0 0 0 0x00000000
0xFA4 0 0 0x00000000
0xFA8 0 0 0x00000000
0xFAC 0 0 0x00000000
0xFB0 0 0 0x00000000
0xFB4 0 0 0x00000000
0xFB8 0 0 0x00000000
0xFBC 0 0 0x00000000
0xFC0 0 0 0x00000000
0xFC4 0 0 0x00000000
0xFC8 0 0 0x00000000
0xFCC 0 0 0x00000000
0xFD0 0 0 0x00000000
0xFD4 0 0 0x00000000
0xFD8 0 0 0x00000000
0xFDC 0 0 0x00000000
0xFE0 0 0 0x00000000
0xFE4 0 0 0x00000000
0xFE8 0 0 0x00000000
0xFEC 0 0 0x00000000
0xFF0 4 4 0x00000004
0xFF4 3 3 0x00000003
0xFF8 2 2 0x00000002
0xFFC 1 1 0x00000001
Type here to search
Mưa 3 cm vào Thứ Sáu
4:35 PM
6/10/2025
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Search
ENG
4:35 PM
6/10/2025

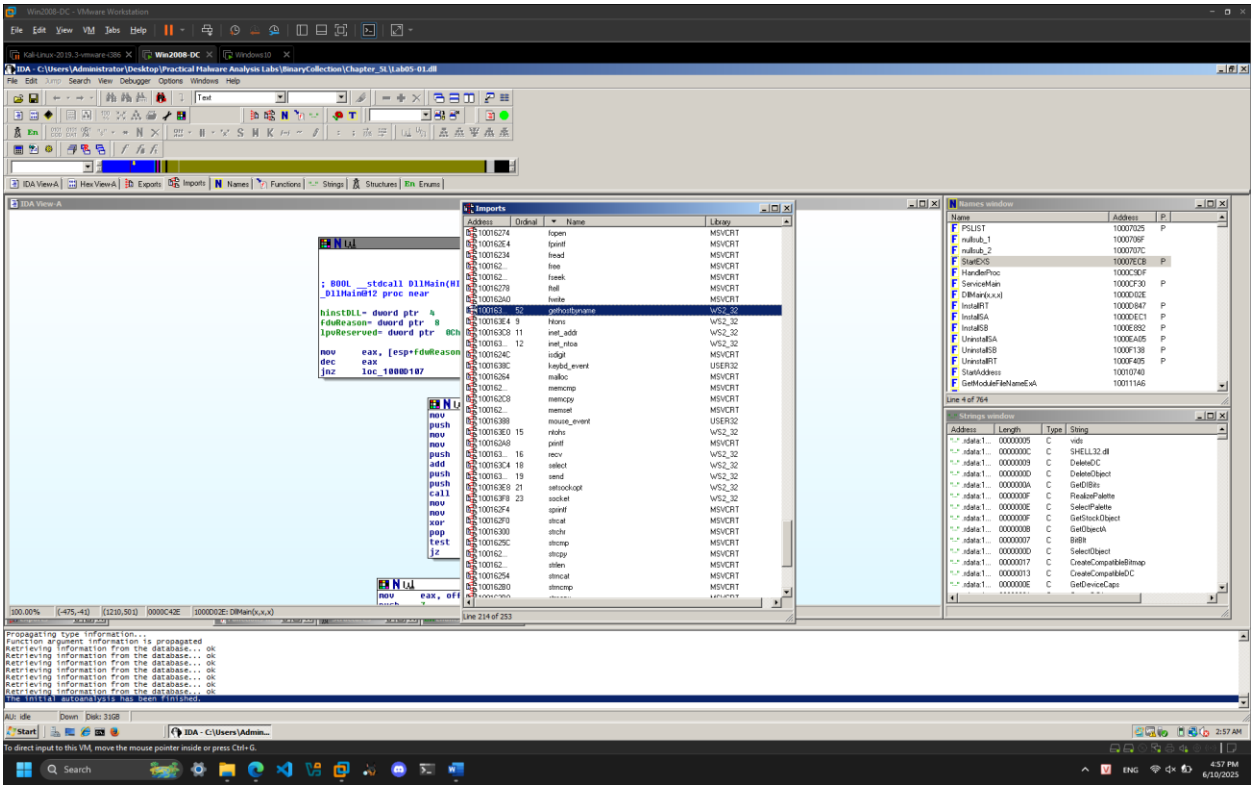

Jasmin - new document
File Edit Run
Welcome
new document
new document
new document
Registers
bin ±dec dec hex
EAX: 4
EBX: 3
ECX: 2
EDX: 1
ESI: 0
EDI: 0
ESP: 4096
EBP: 4096
EIP: 12
Carry Overflow
Sign Zero
Parity Auxiliary
Trap Direction
FPU Registers
name value
ST0 0.0
ST1 0.0
ST2 0.0
ST3 0.0
ST4 0.0
ST5 0.0
ST6 0.0
ST7 0.0
0 mov eax, 1
1 mov ebx, 2
2 mov ecx, 3
3 mov edx, 4
4 push eax
5 push ebx
6 push ecx
7 push edx
8 pop eax
9 pop ebx
10 pop ecx
11 pop edx
Help
7-Segment
StripLight
Console
Graphics
Command: POP-Pop a Word from the Stack
Arguments: DEST 16 or 32 bit memory location or register or segment register
Usage: POP DEST
Effects: POP stores the contents of DEST with the word on the top of the 80386 stack, addressed by SS:SP (address-size attribute of 16 bits) or SS:ESP (addresssize attribute of 32 bits). The stack pointer is then increased accordingly.
Flags to be set: none
Memory
desc hex highlight
8 Bit 16Bit 32Bit
address signed int unsigned int hex
0xF34 0 0 0x00000000
0xF38 0 0 0x00000000
0xF3C 0 0 0x00000000
0xF40 0 0 0x00000000
0xF44 0 0 0x00000000
0xF48 0 0 0x00000000
0xF4C 0 0 0x00000000
0xF50 0 0 0x00000000
0xF54 0 0 0x00000000
0xF58 0 0 0x00000000
0xF5C 0 0 0x00000000
0xF60 0 0 0x00000000
0xF64 0 0 0x00000000
0xF68 0 0 0x00000000
0xF6C 0 0 0x00000000
0xF70 0 0 0x00000000
0xF74 0 0 0x00000000
0xF78 0 0 0x00000000
0xF7C 0 0 0x00000000
0xF80 0 0 0x00000000
0xF84 0 0 0x00000000
0xF88 0 0 0x00000000
0xF8C 0 0 0x00000000
0xF90 0 0 0x00000000
0xF94 0 0 0x00000000
0xF98 0 0 0x00000000
0xF9C 0 0 0x00000000
0xFA0 0 0 0x00000000
0xFA4 0 0 0x00000000
0xFA8 0 0 0x00000000
0xFAC 0 0 0x00000000
0xFB0 0 0 0x00000000
0xFB4 0 0 0x00000000
0xFB8 0 0 0x00000000
0xFBC 0 0 0x00000000
0xFC0 0 0 0x00000000
0xFC4 0 0 0x00000000
0xFC8 0 0 0x00000000
0xFCC 0 0 0x00000000
0xFD0 0 0 0x00000000
0xFD4 0 0 0x00000000
0xFD8 0 0 0x00000000
0xFDC 0 0 0x00000000
0xFE0 0 0 0x00000000
0xFE4 0 0 0x00000000
0xFE8 0 0 0x00000000
0xFEC 0 0 0x00000000
0xFF0 4 4 0x00000004
0xFF4 3 3 0x00000003
0xFF8 2 2 0x00000002
0xFFC 1 1 0x00000001
Type here to search
4:36 PM
6/10/2025
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Search
ENG
4:36 PM
6/10/2025

2. IDA Pro

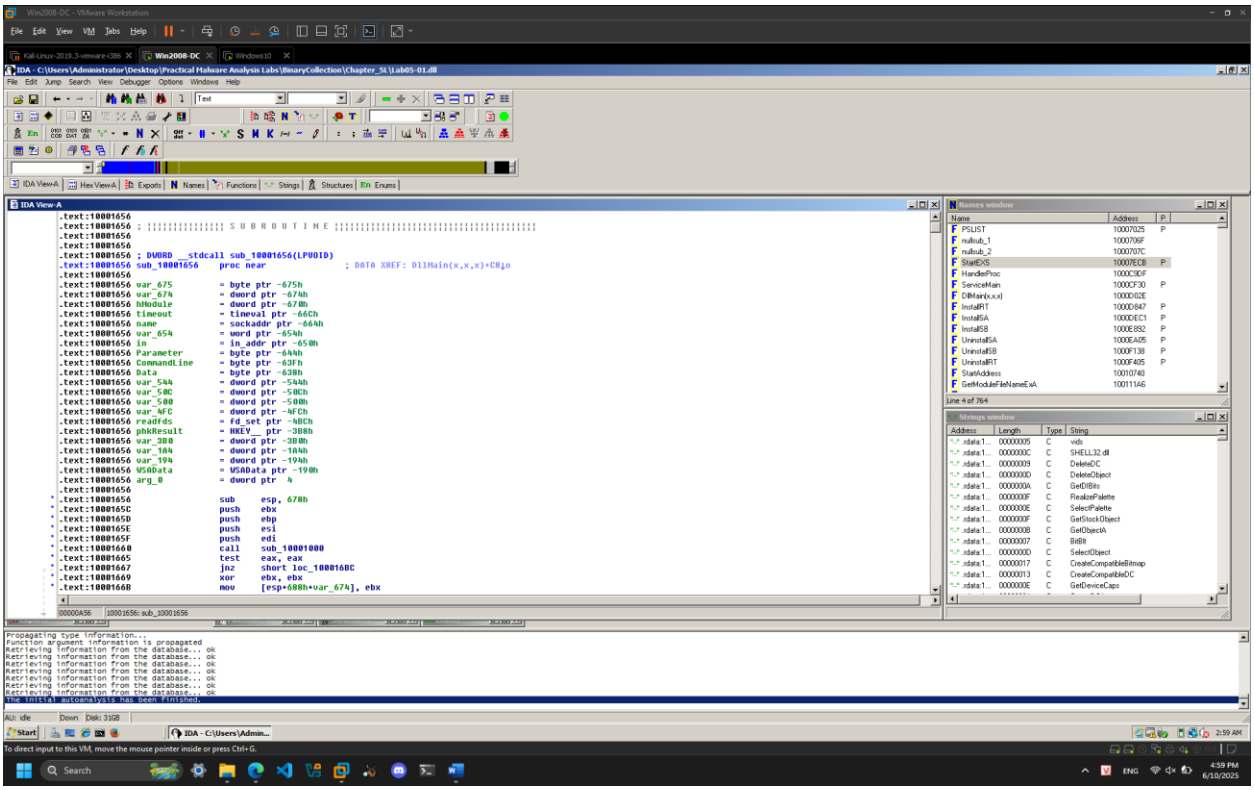
Finding the Address of DLLMain:



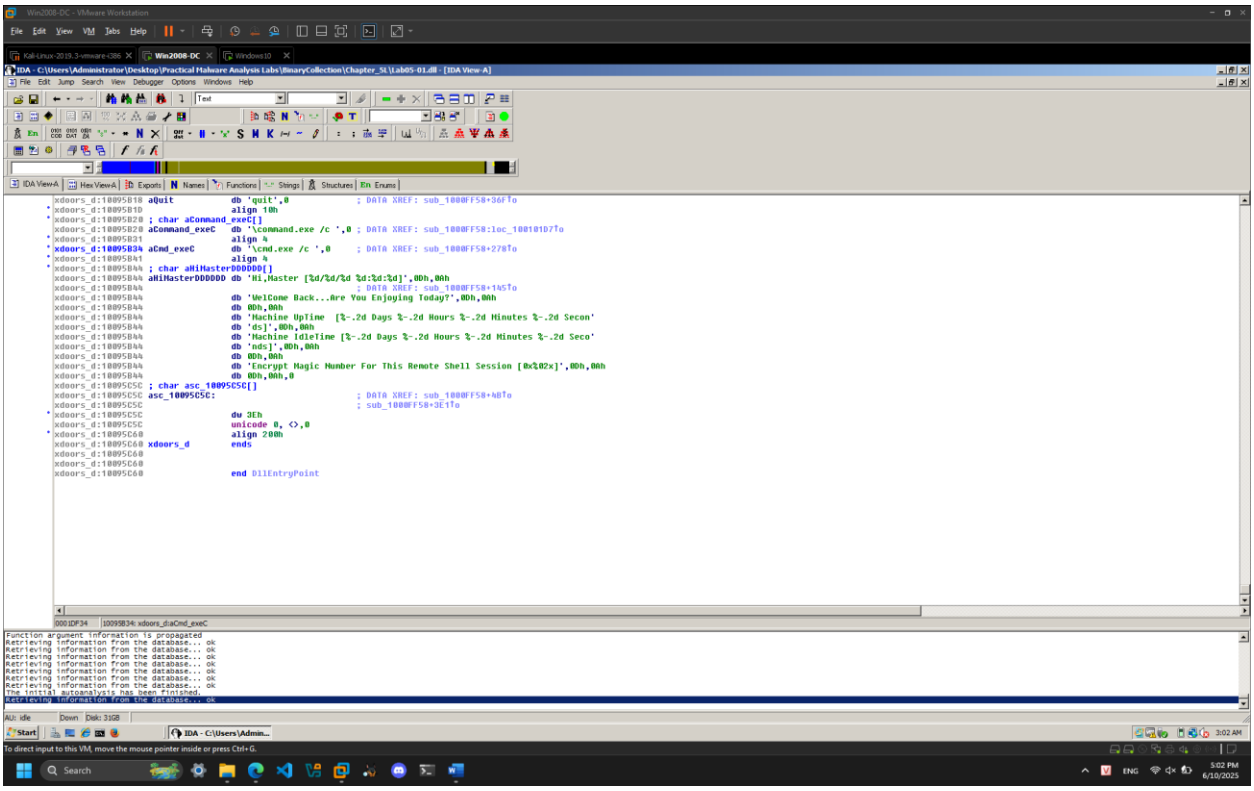
Find the import for gethostbyname:

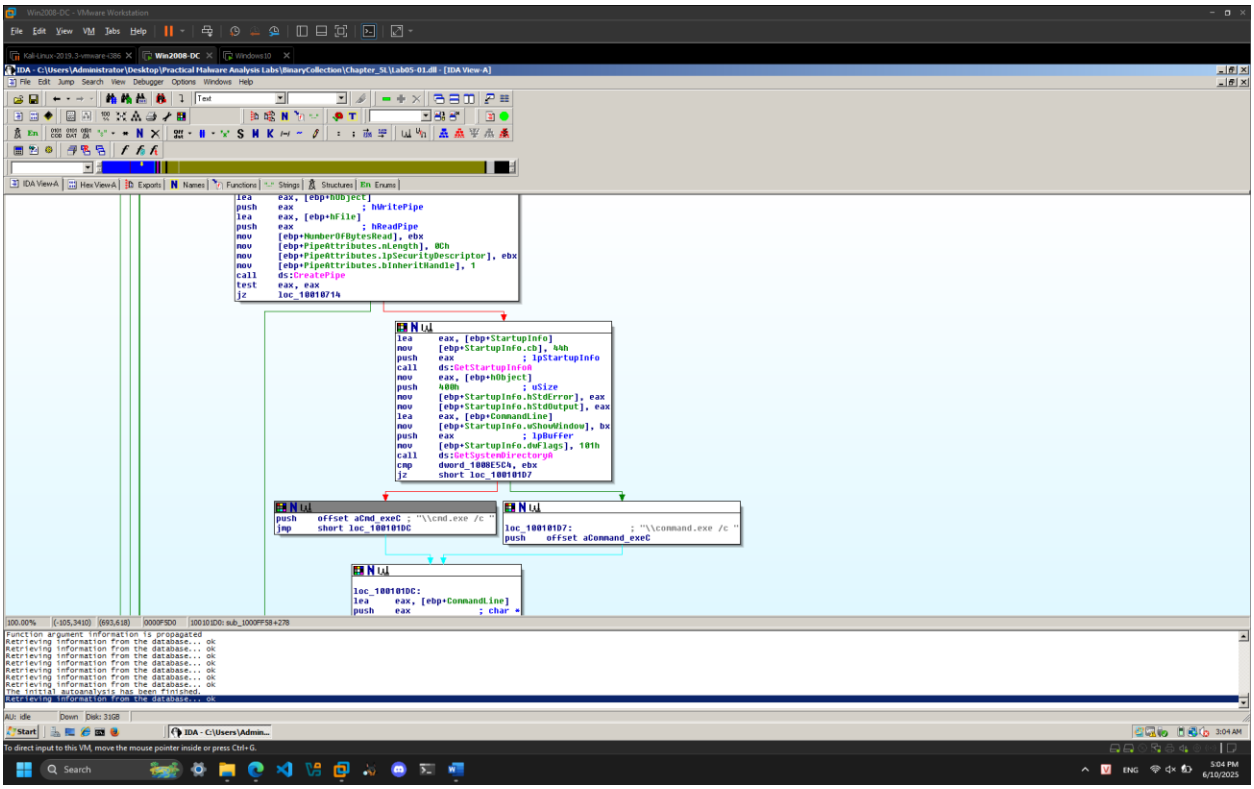


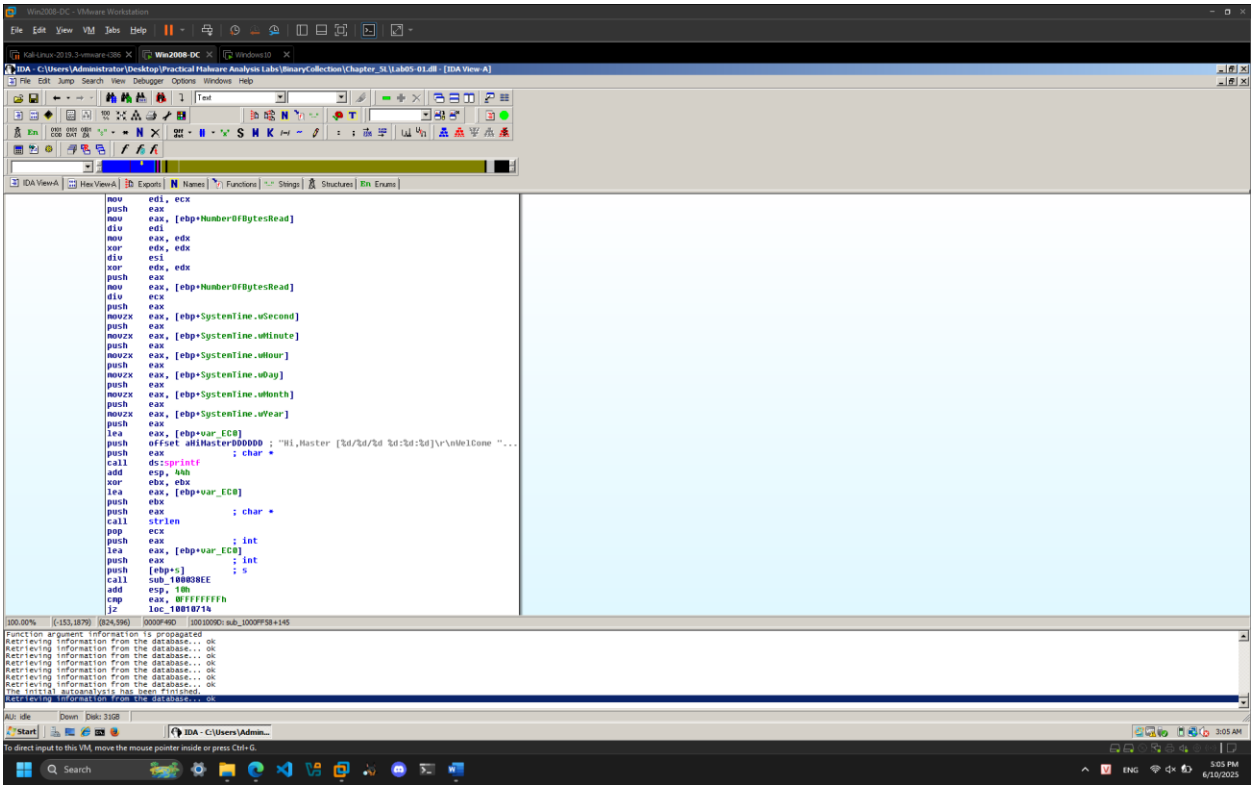
Count Local Variables for the Subroutine at 0x10001656:

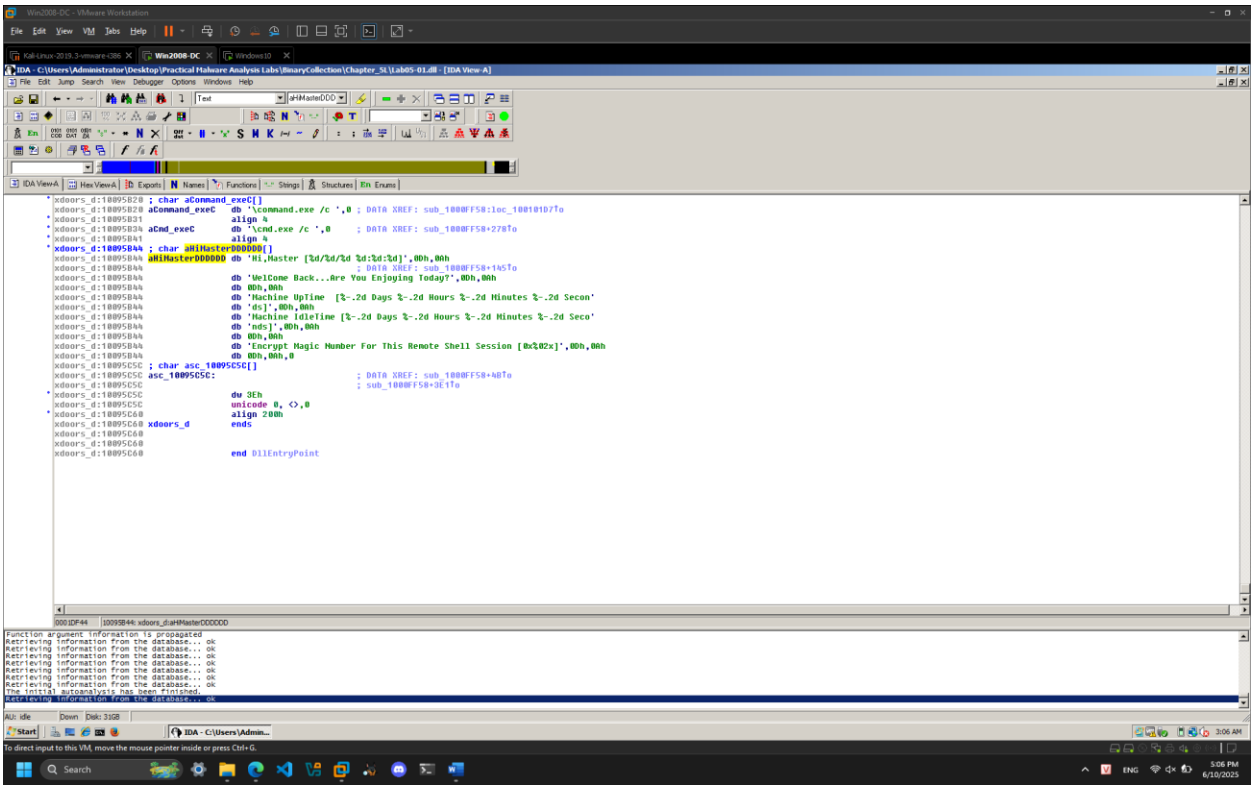


Finding the Purpose of the Code that References \cmd.exe /c:



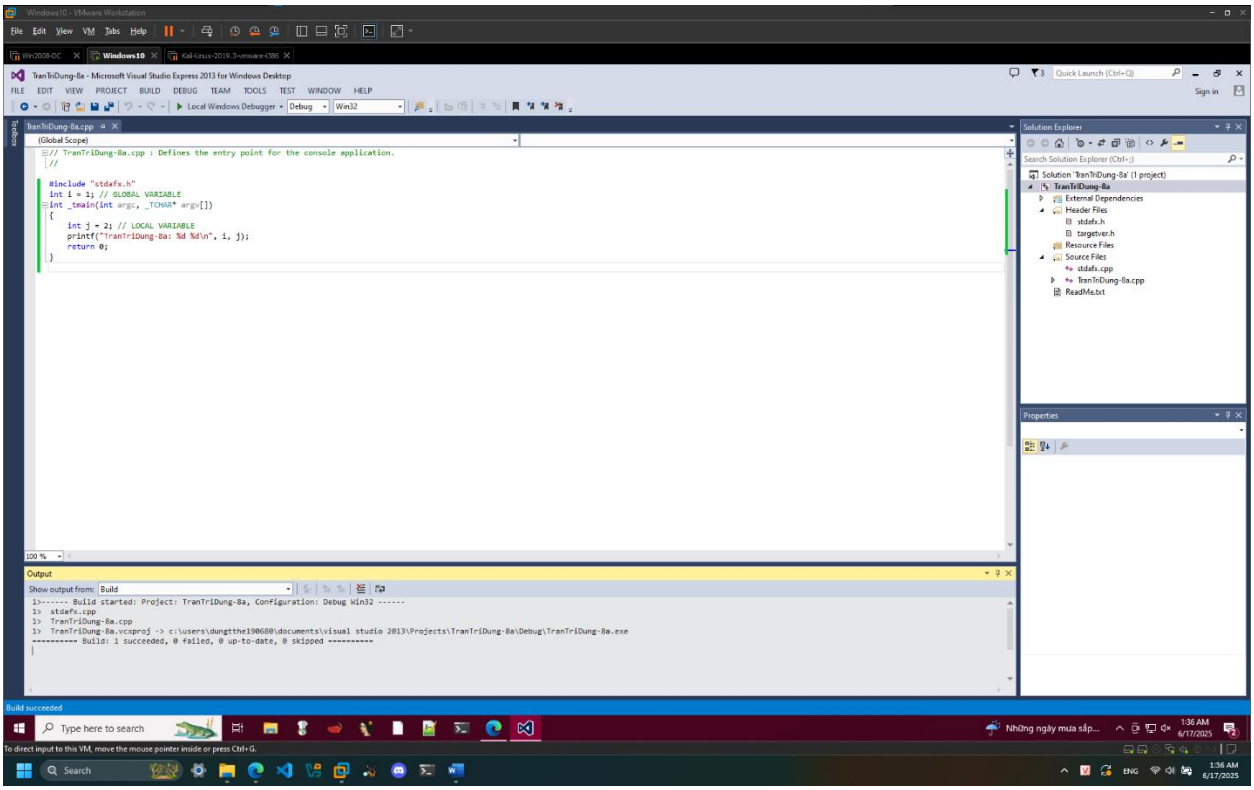




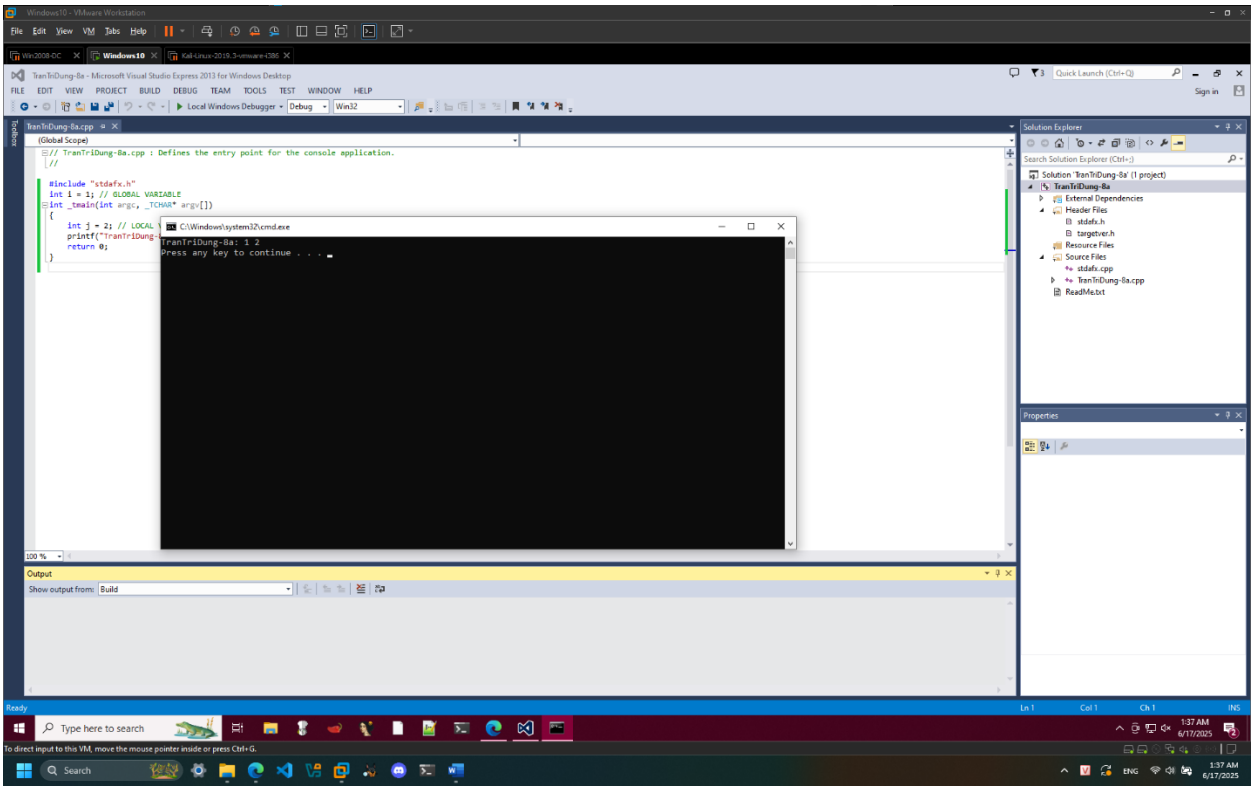


3. Disassembling C on Windows

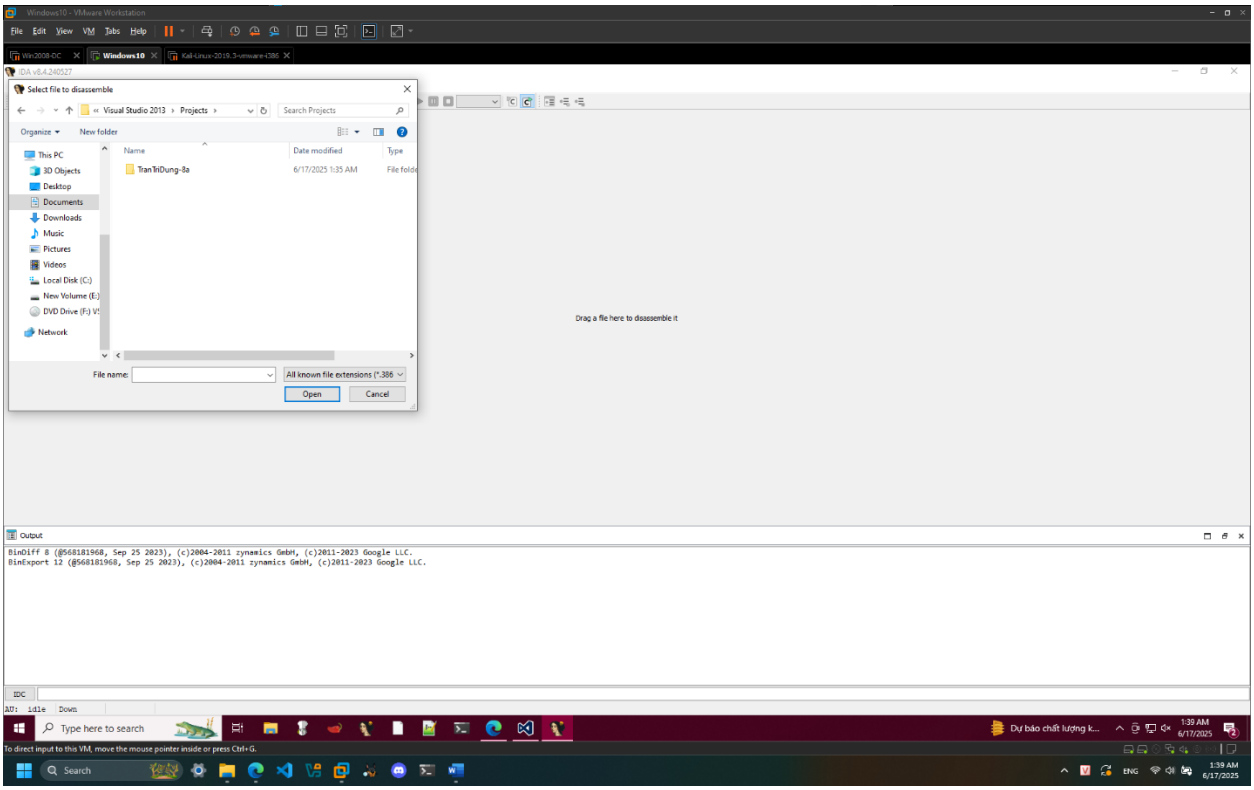
Global and Local Variables:

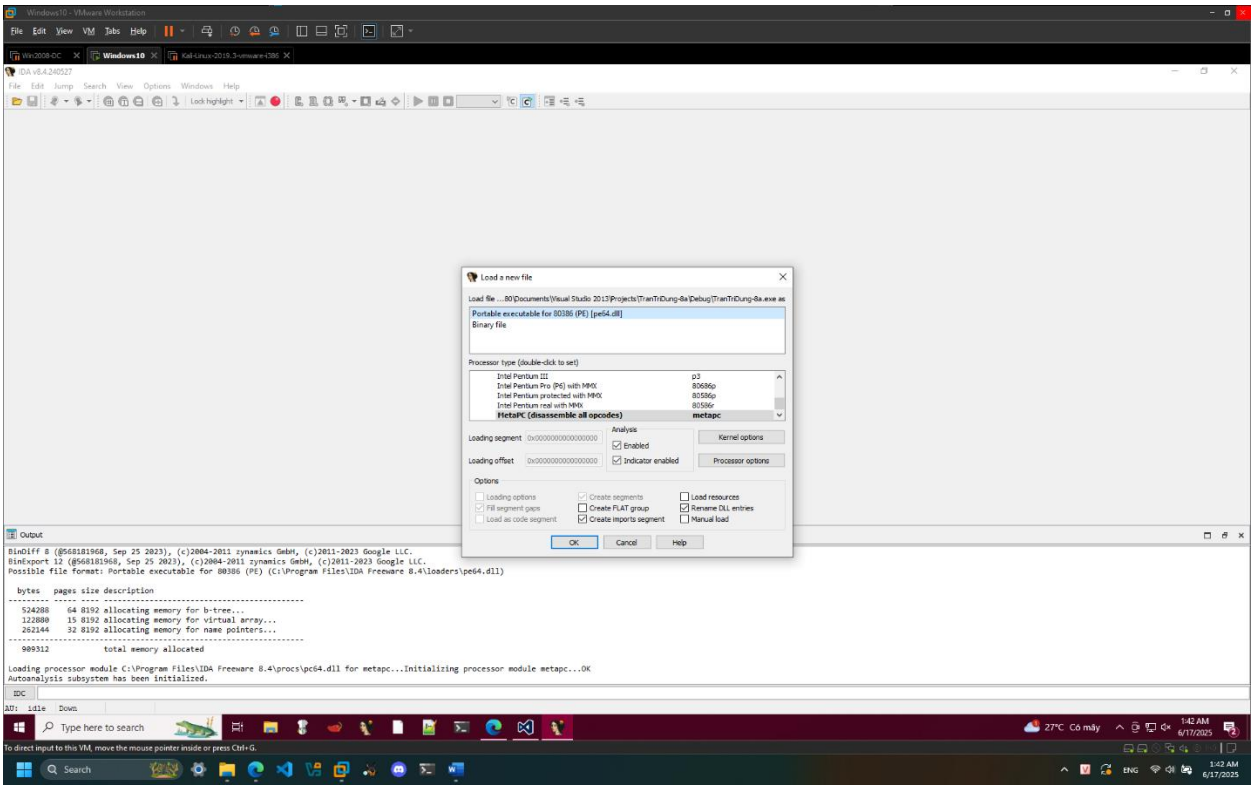


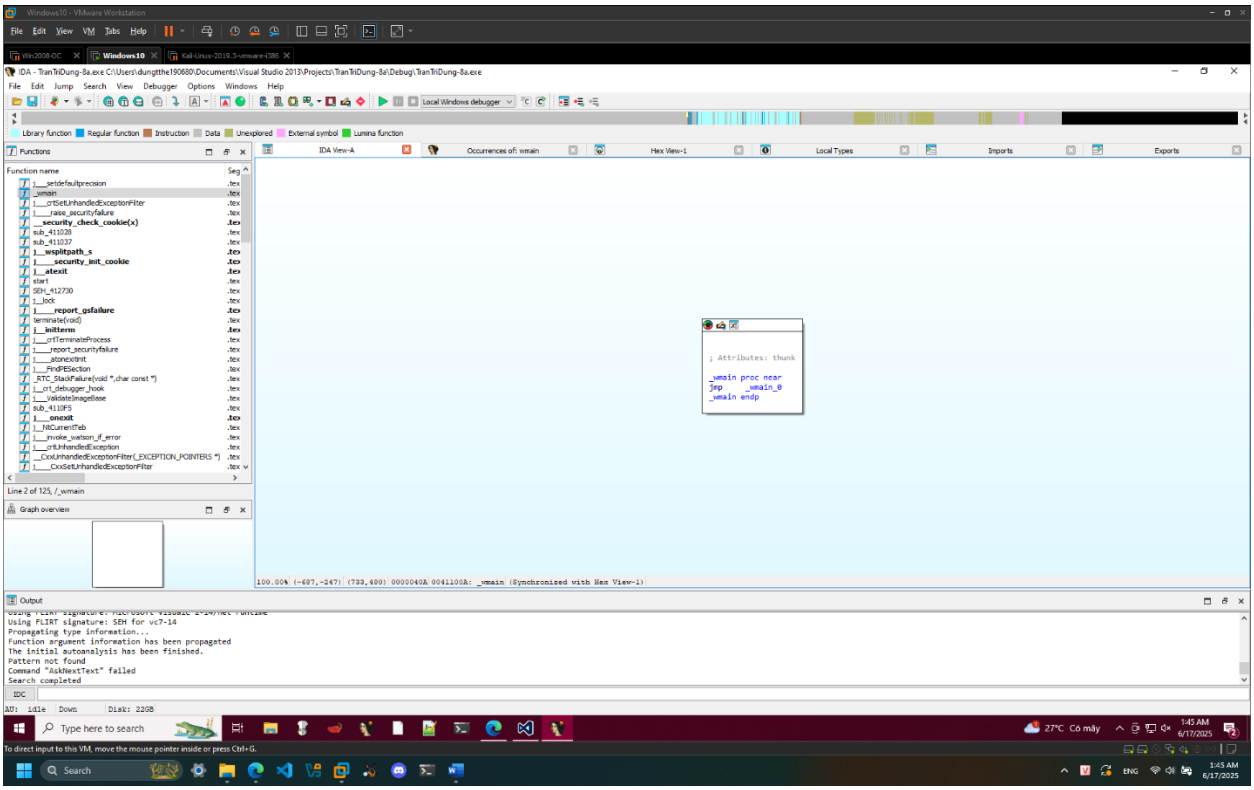
Compiling and Running Program:

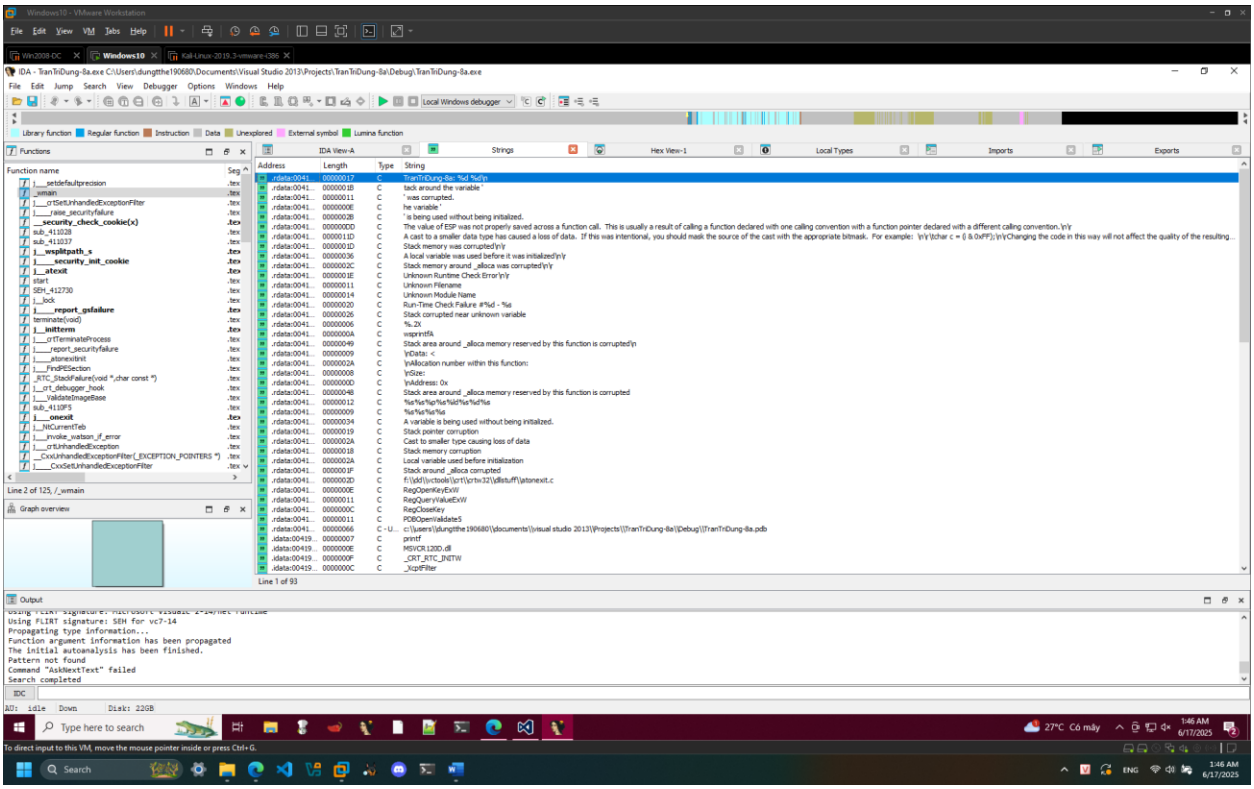


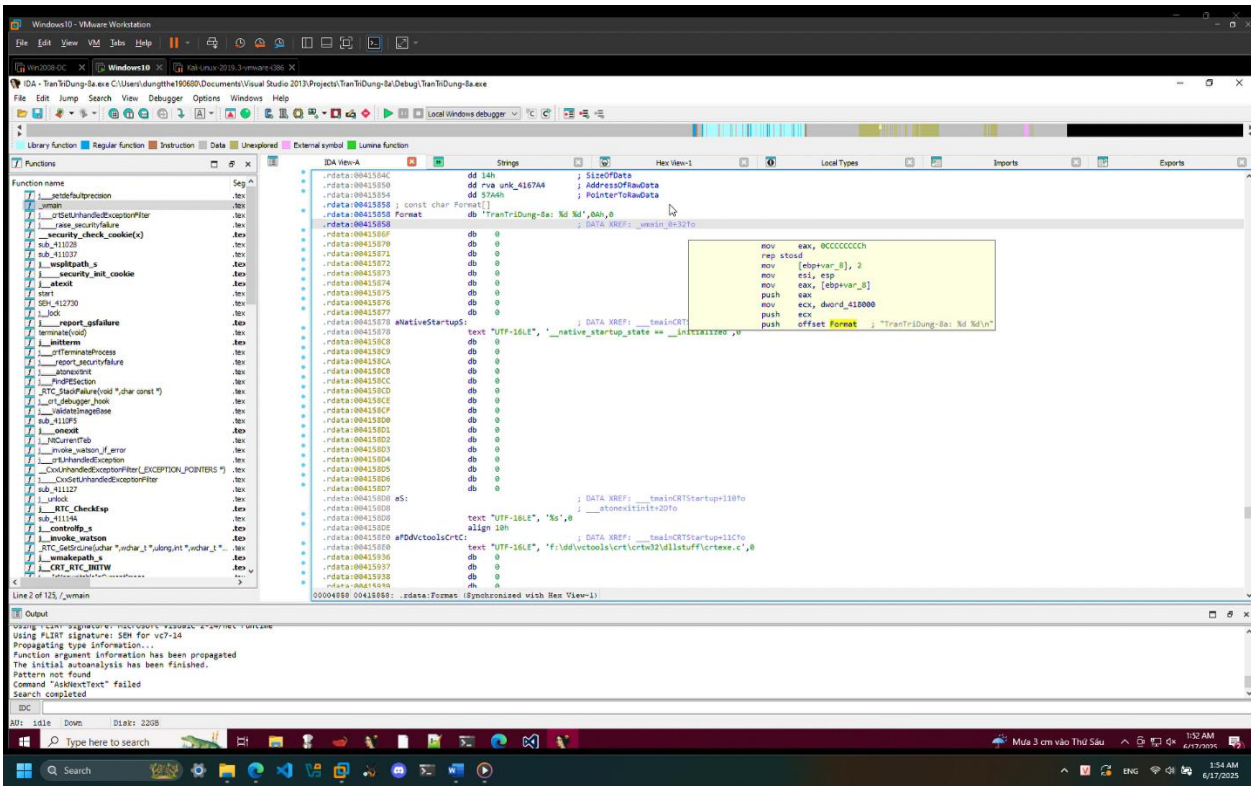
Disassembling the EXE:

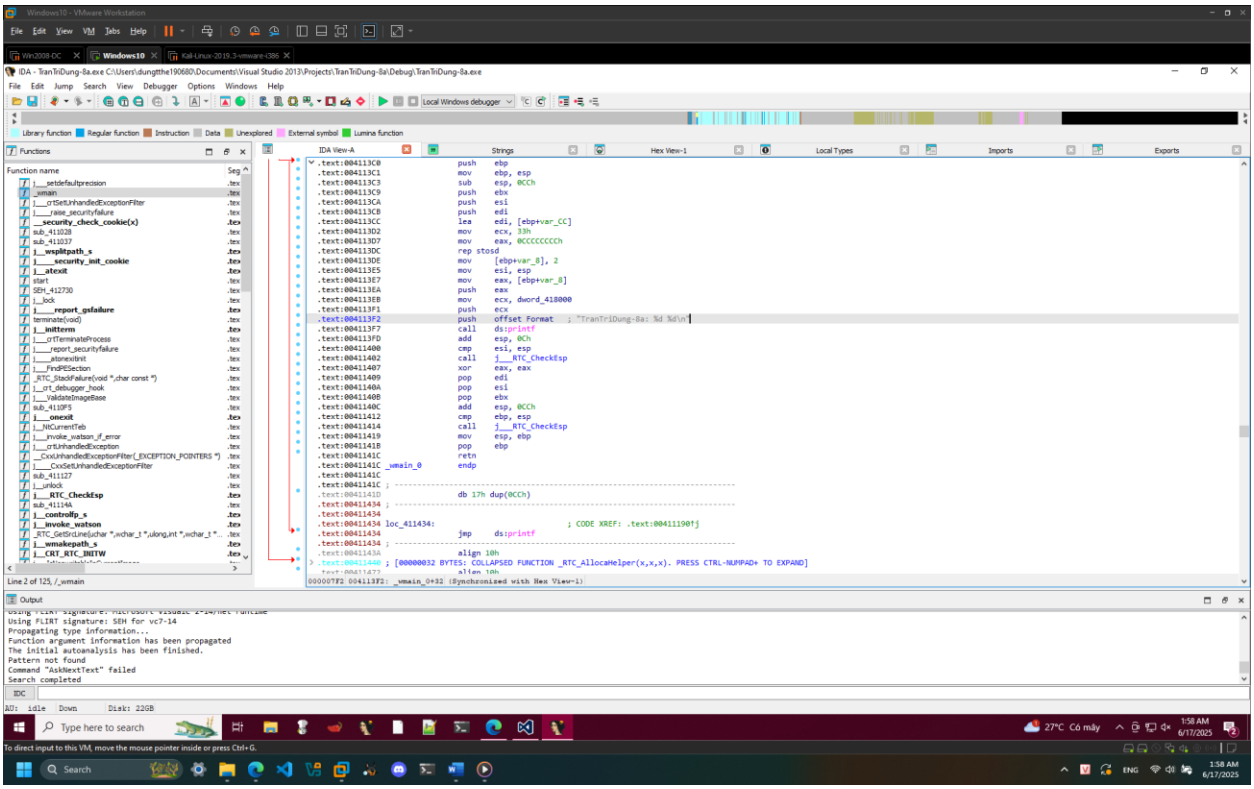



Load a new file
Portable executable for 80386 (PE) [pe64.dll]
Binary file
Processor type (double-click to set)
Intel Pentium III p3
Intel Pentium Pro (P6) with MMX 80686p
Intel Pentium protected with MMX 80586p
Intel Pentium real with MMX 80586r
MetaPC (disassemble all opcodes) metapc
Loading segment
Loading offset
Analysis
Enabled
Indicator enabled
Kernel options
Processor options
Options
Loading options
Fill segment gaps
Load as code segment
Create segments
Create FLAT group
Create imports segment
Load resources
Rename DLL entries
Manual load
OK
Cancel
Help
Output
BinDiff 8 (@568181968, Sep 25 2023), (c)2004-2011 zynamics GmbH, (c)2011-2023 Google LLC.
BinExport 12 (@568181968, Sep 25 2023), (c)2004-2011 zynamics GmbH, (c)2011-2023 Google LLC.
Possible file format: Portable executable for 80386 (PE) (C:\Program Files\IDA Freeware 8.4\loaders\pe64.dll)
bytes pages size description
524288 64 8192 allocating memory for b-tree...
122880 15 8192 allocating memory for virtual array...
262144 32 8192 allocating memory for name pointers...
909312 total memory allocated
Loading processor module C:\Program Files\IDA Freeware 8.4\procs\pc64.dll for metapc...Initializing processor module metapc...OK
Autoanalysis subsystem has been initialized.
IDC
AU: idle Down
Type here to search
27°C Có mây
1:42 AM
6/17/2025
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Search
ENG

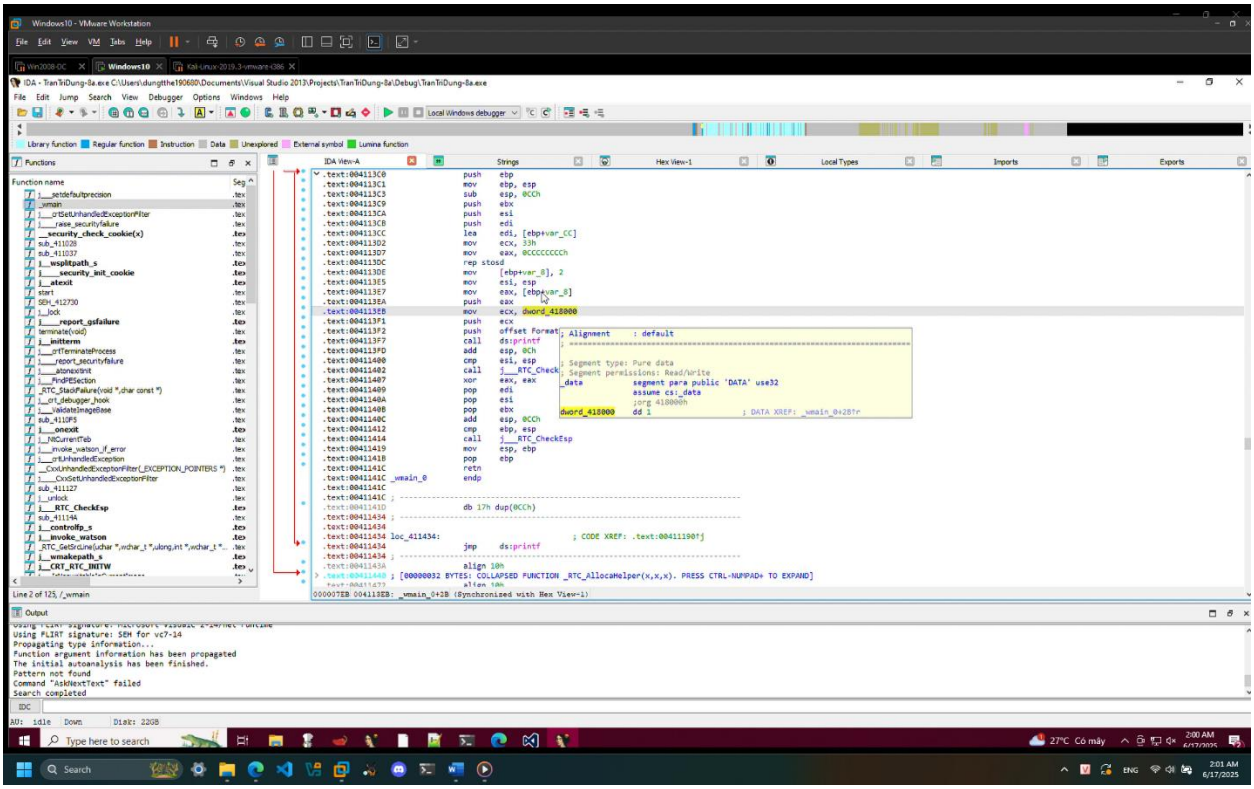

IDA - TranTriDung-8a.exe C:\Users\dungtthe190680\Documents\Visual Studio 2013\Projects\TranTriDung-8a\Debug\TranTriDung-8a.exe
Functions
IDA View-A
Occurrences of: wmain
Hex View-1
Local Types
Imports
Exports
Graph overview
; Attributes: thunk
_wmain proc near
jmp __wmain_0
_wmain endp
Output
Using FLIRT signature: SEH for vc7-14
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Pattern not found
Command "AskNextText" failed
Search completed
IDC
AU: idle Down Disk: 22GB
Type here to search
27°C Có mây
1:45 AM
6/17/2025
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Search
ENG



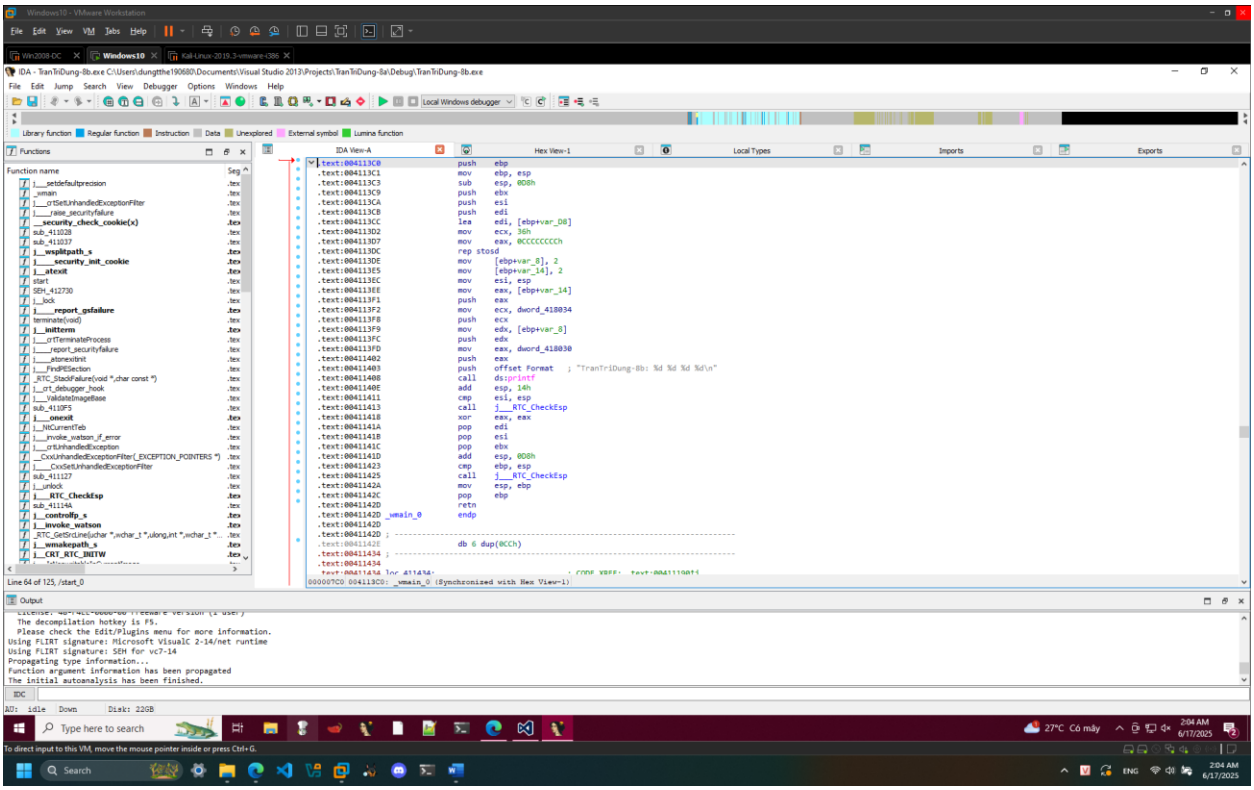




Understanding Global and Local Variables:

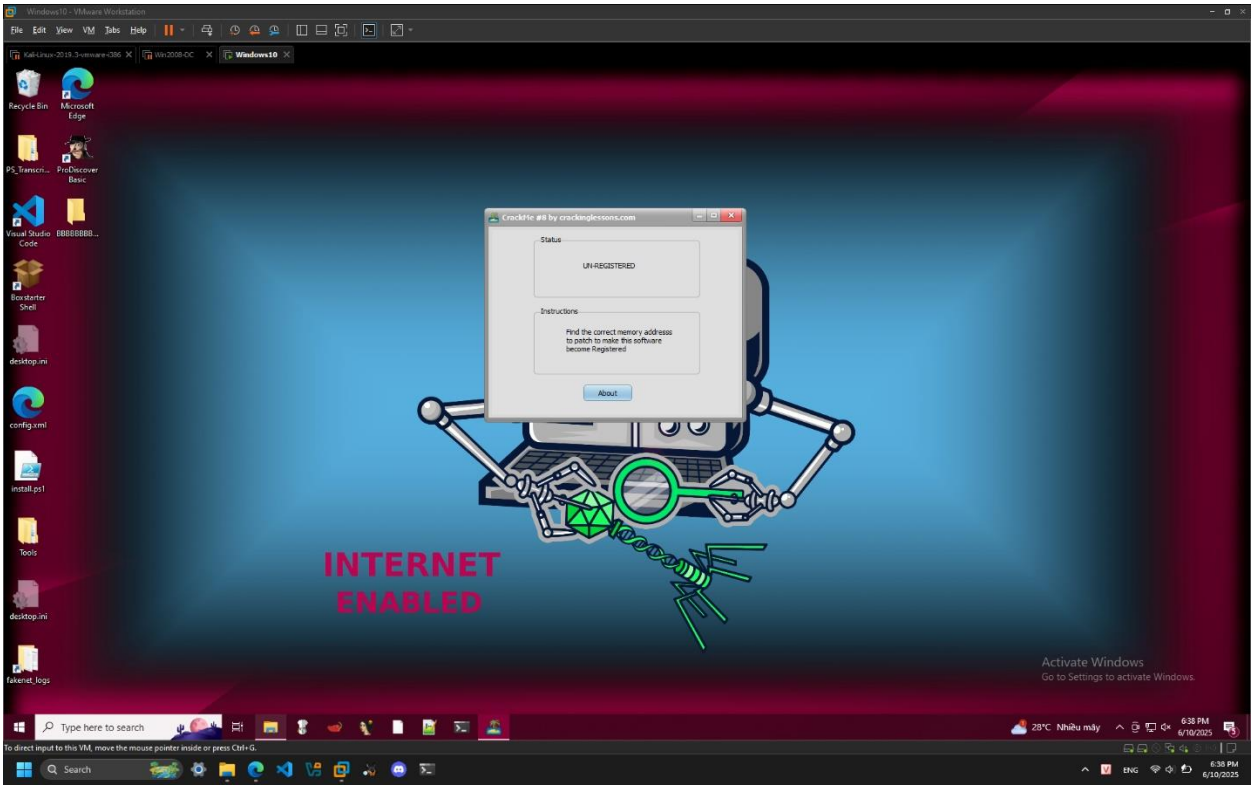


Extra Credit:

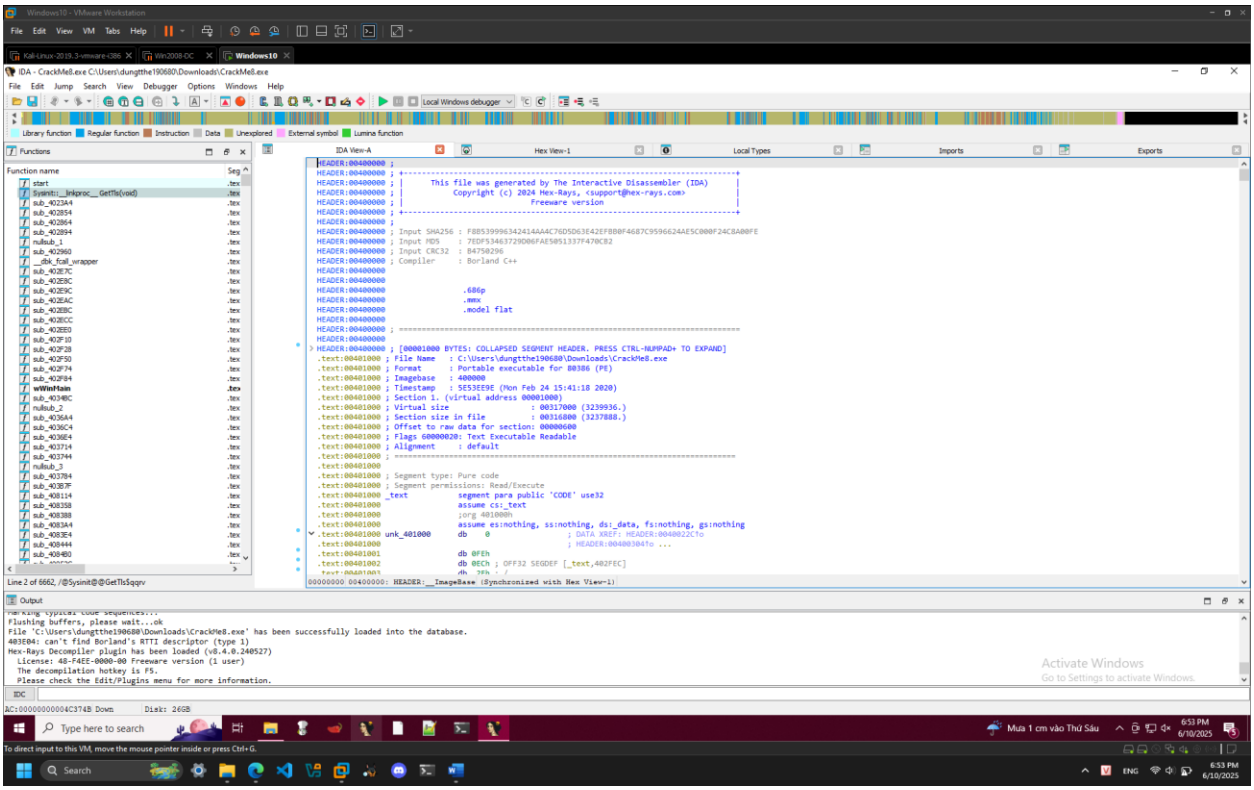


CrackMe #8

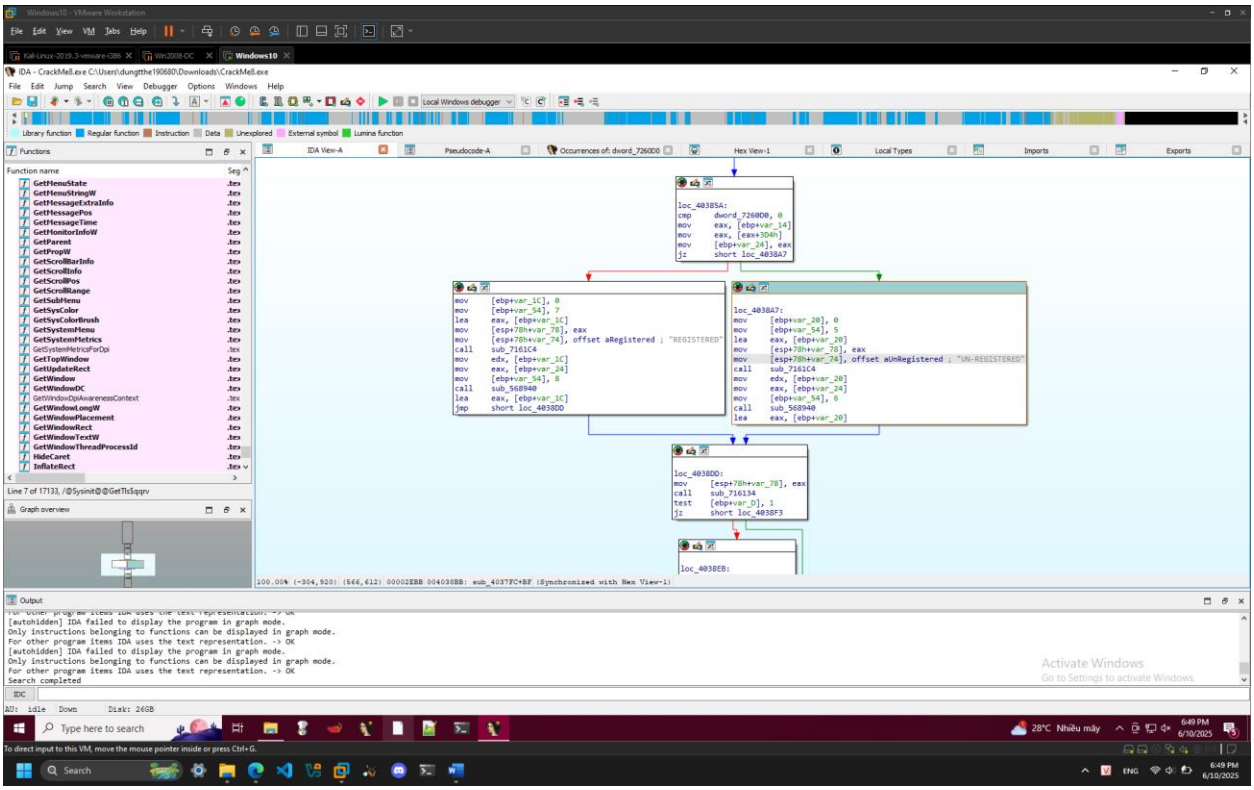
Đầu tiên em thử chạy chương trình và nhận được thông báo có từ khóa “UN-REGISTERED”:



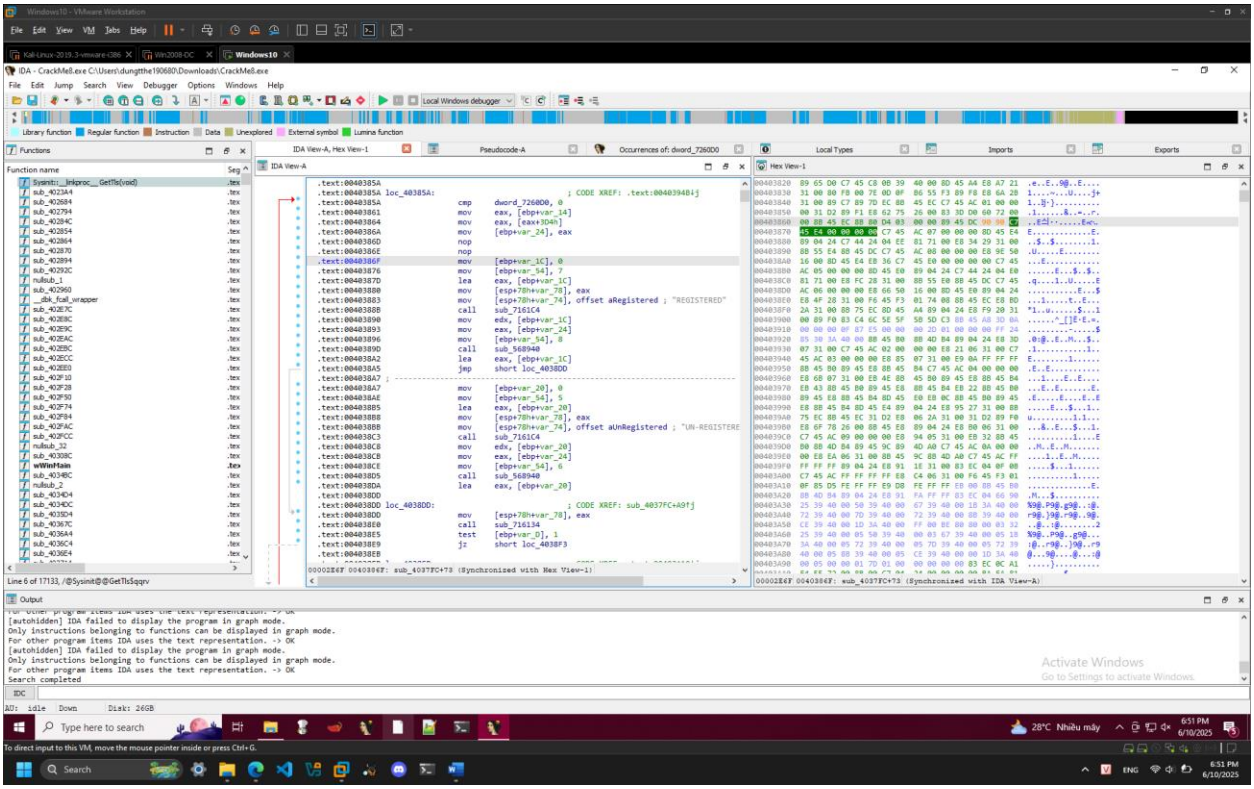
Sau đó em mở phần mềm IDA và tiến hành dịch ngược file:



Em search từ khóa “UN-REGISTERED” và được điều hướng đến đoạn code này:



Có thể thấy thông báo đã đăng ký không được hiển thị do lệnh nhảy trong đoạn loc_40385A, vì vậy để hiển thị thông báo đã đăng ký, em xóa lệnh nhảy bằng cách sửa mã hex thành 90 (nop):



Lưu lại và chạy chương trình thì em nhận được thông báo đã đăng ký: