**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота № 3**

з дисципліни

«Системне програмування»

на тему

“Дослідження структури програм формату EXE”

Виконав:

Перевірив:

студент групи ІП-93 доцент

Марченко Максим Олегович Павлов В.Г.

номер залікової книжки: 9316

Київ 2020

Мета роботи : ознайомлення із специфікацією COFF. Вивчення прийомів дослідження структури файлу РЕ-формату.

Структура програм формату ЕХЕ:

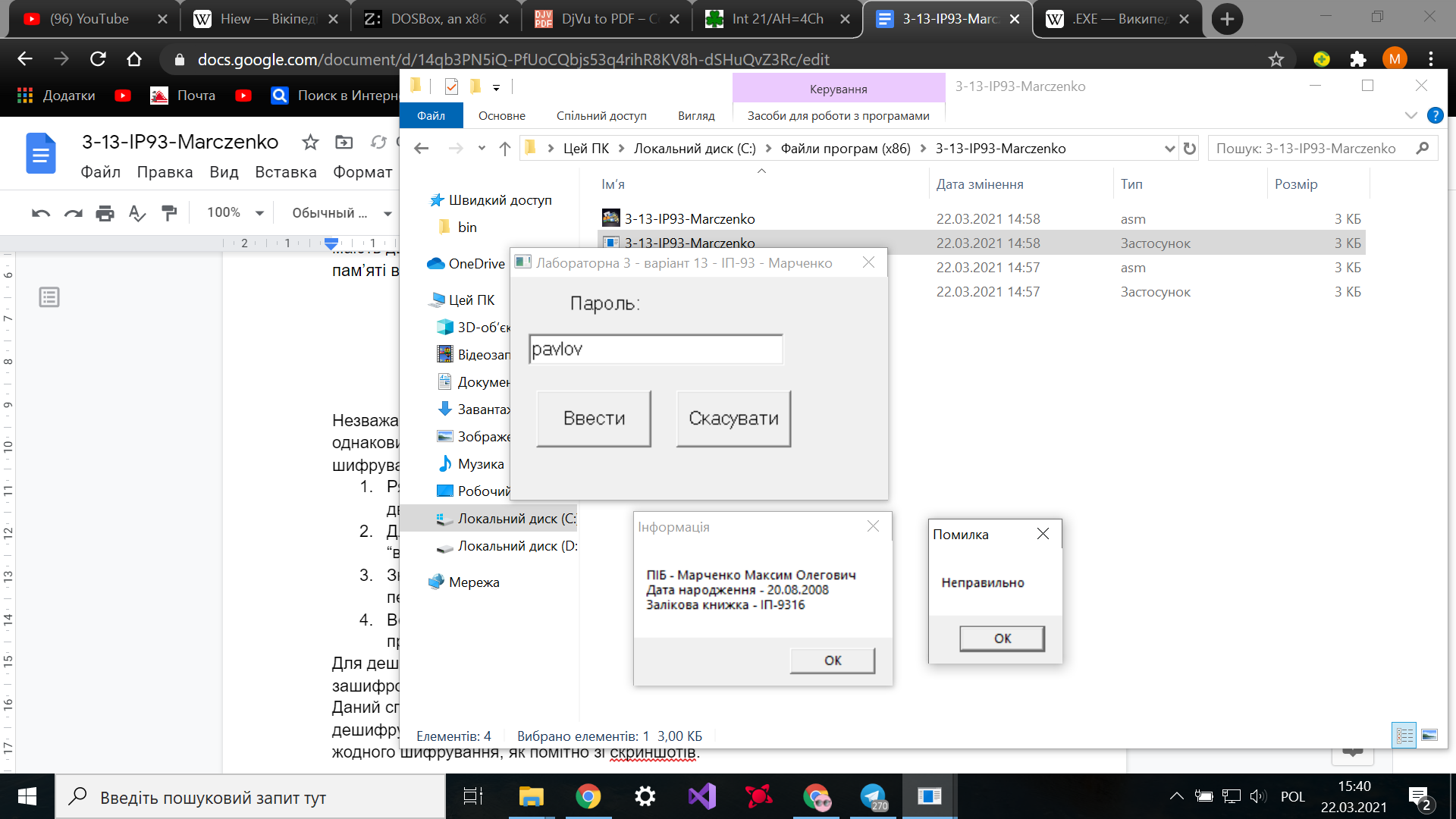
.ехе (executable - виконуваний) - це розширення виконуваного файлу, що застосовується в системах DOS, Microsoft Windows, та інших.

Файл EXE складається з двох частин: заголовок (інформація до завантажувача програми) та модуль завантаження. Заголовок займає перші 25 рядків файлу, та містить, зокрема інформацію про довжину файлу, розмір пам’яті, що потрібно виділити.

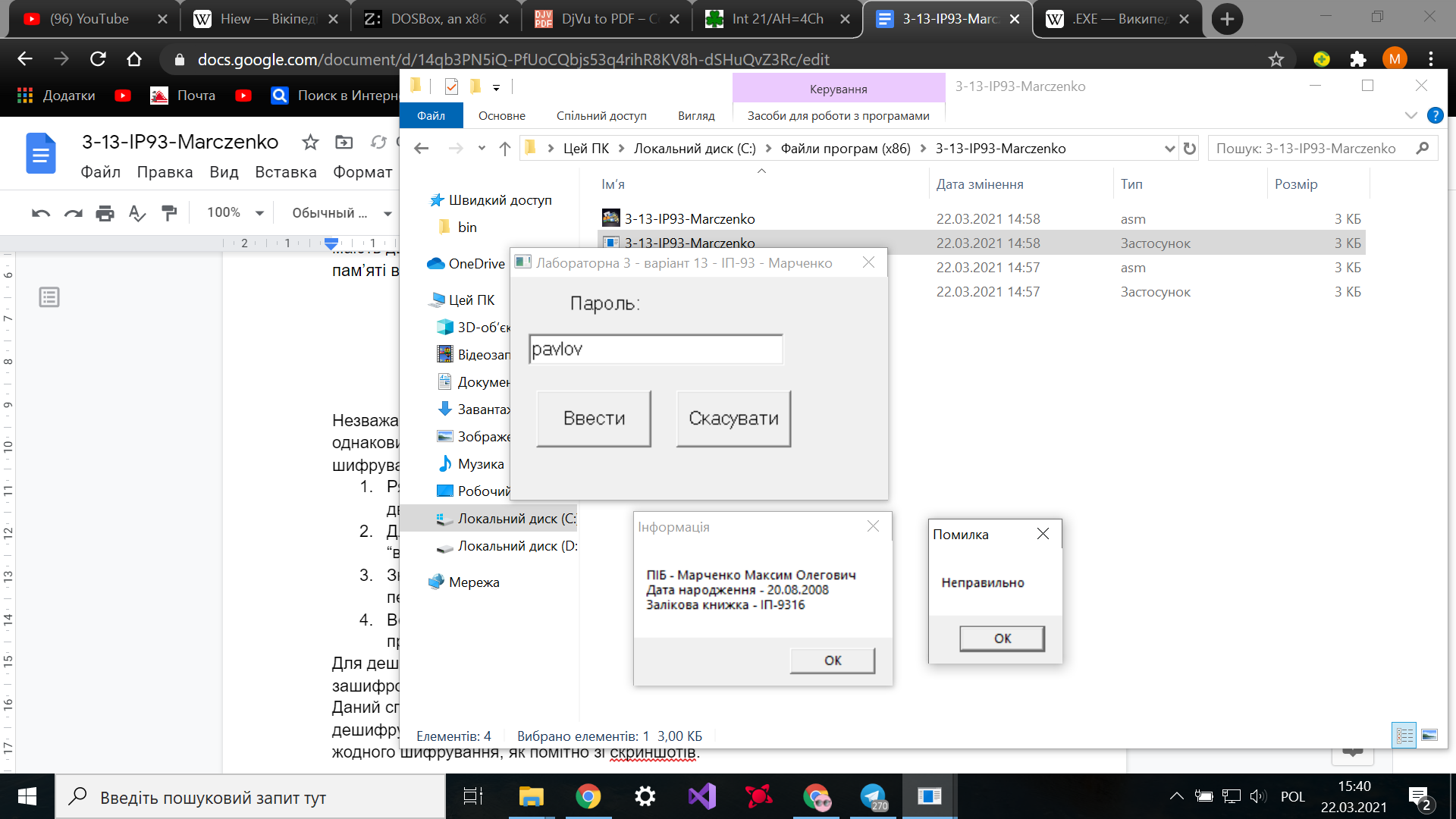
Після нього записана таблиця налаштування адрес, що складається з елементів, що мають два двобайтових поля - зміщення та адресу. що потрібно налаштувати на місце пам’яті в яке завантажується конкретна задача.

Скріншоти виконання програми

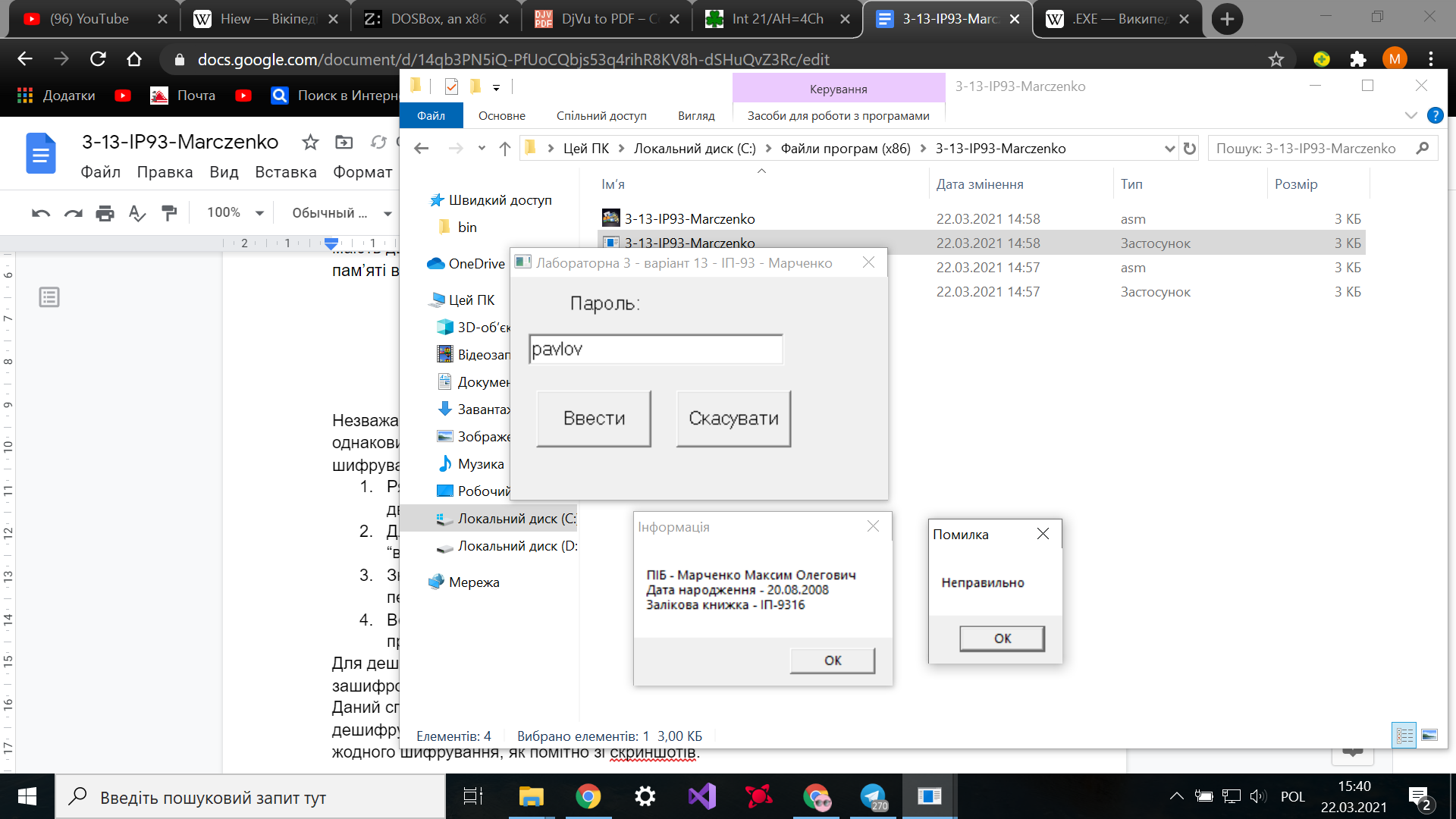
Вікно введення паролю



Після введення правильного паролю

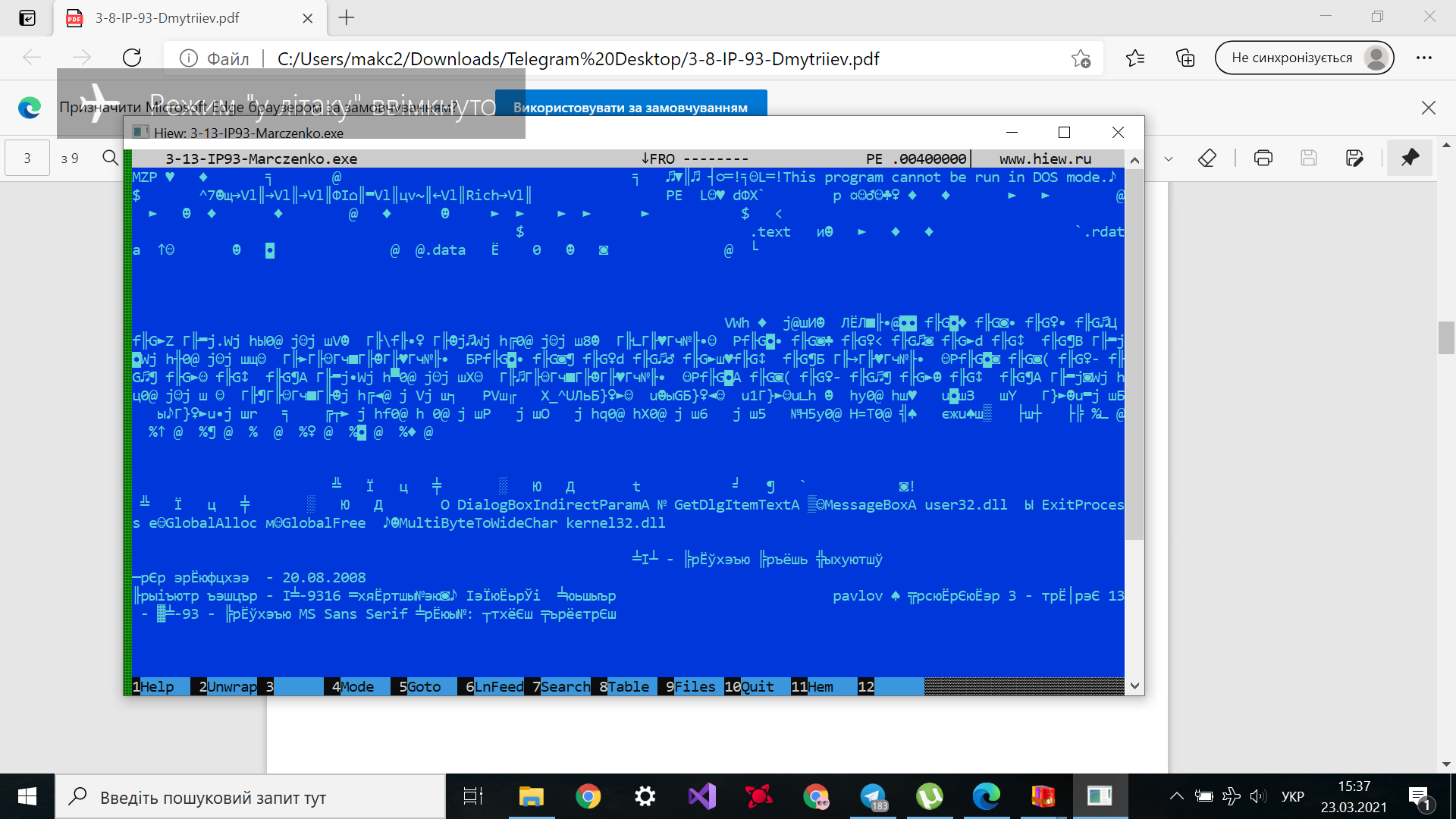


Після введення неправильного паролю

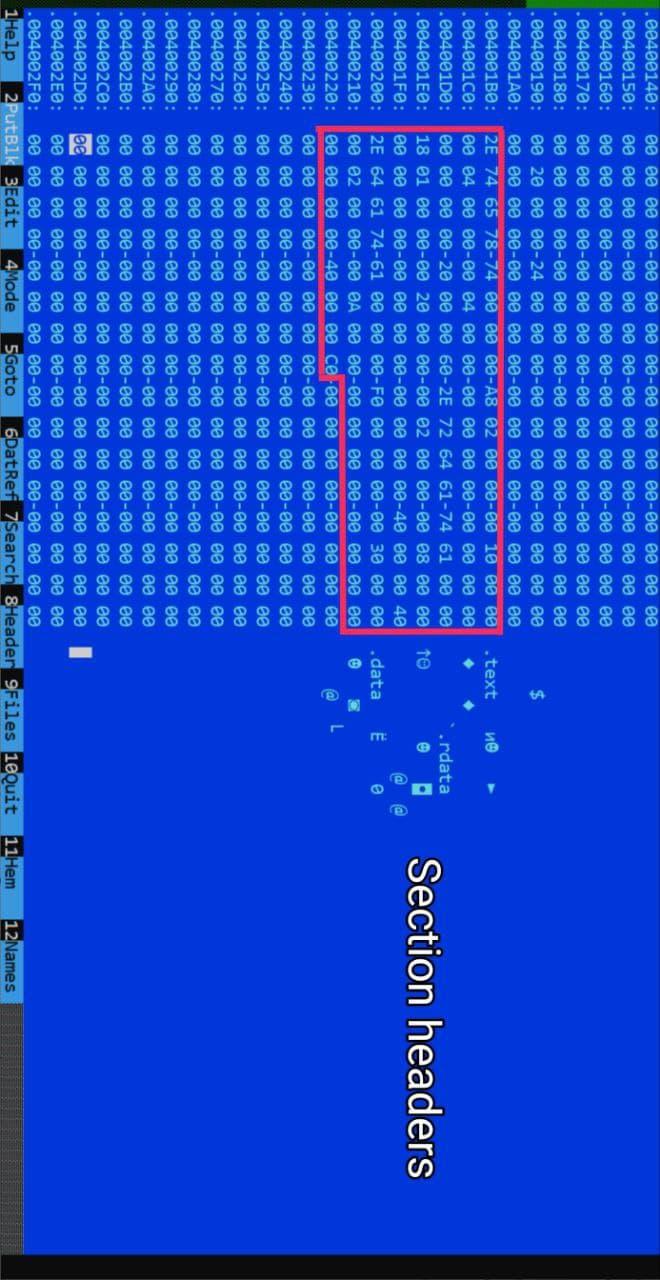
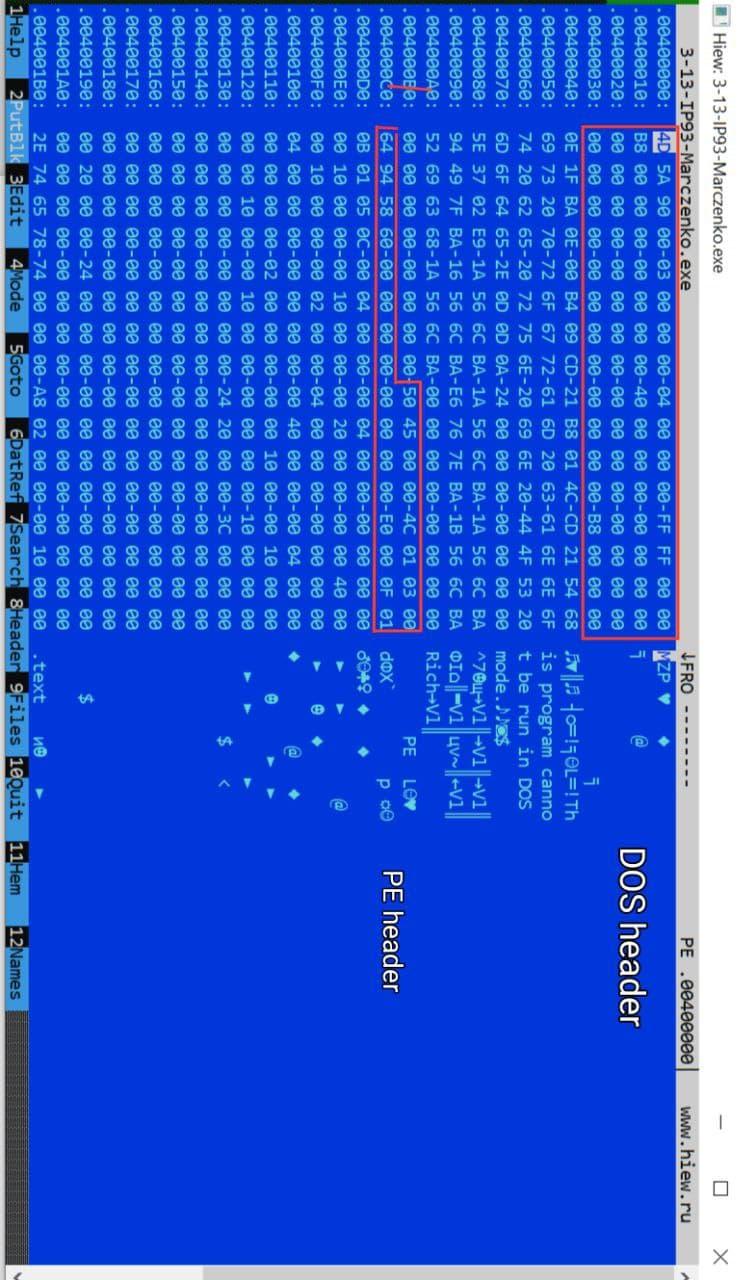


Дослідження файлу в HIEW32

Текстовий формат



Шістнадцятковий формат

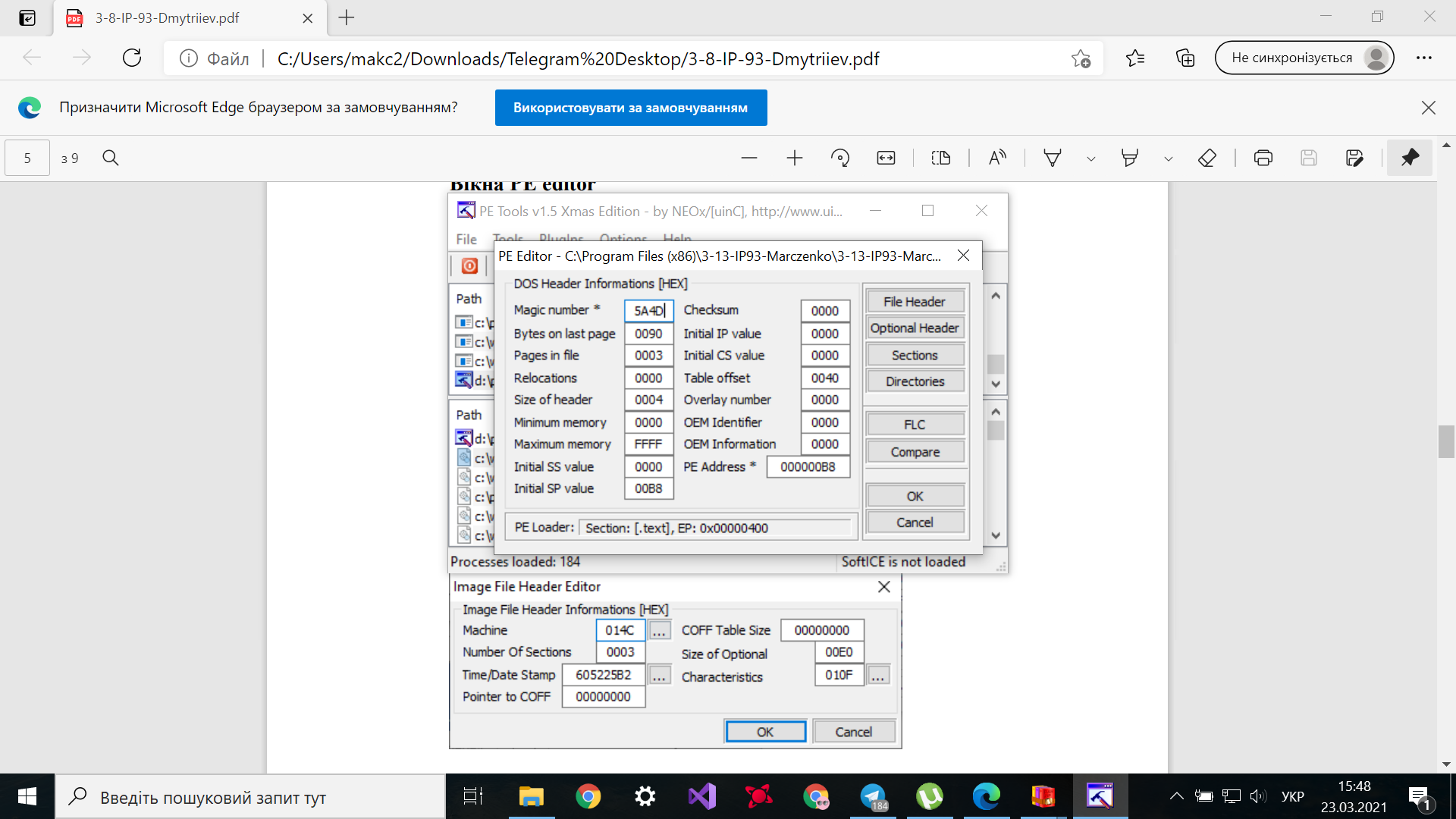


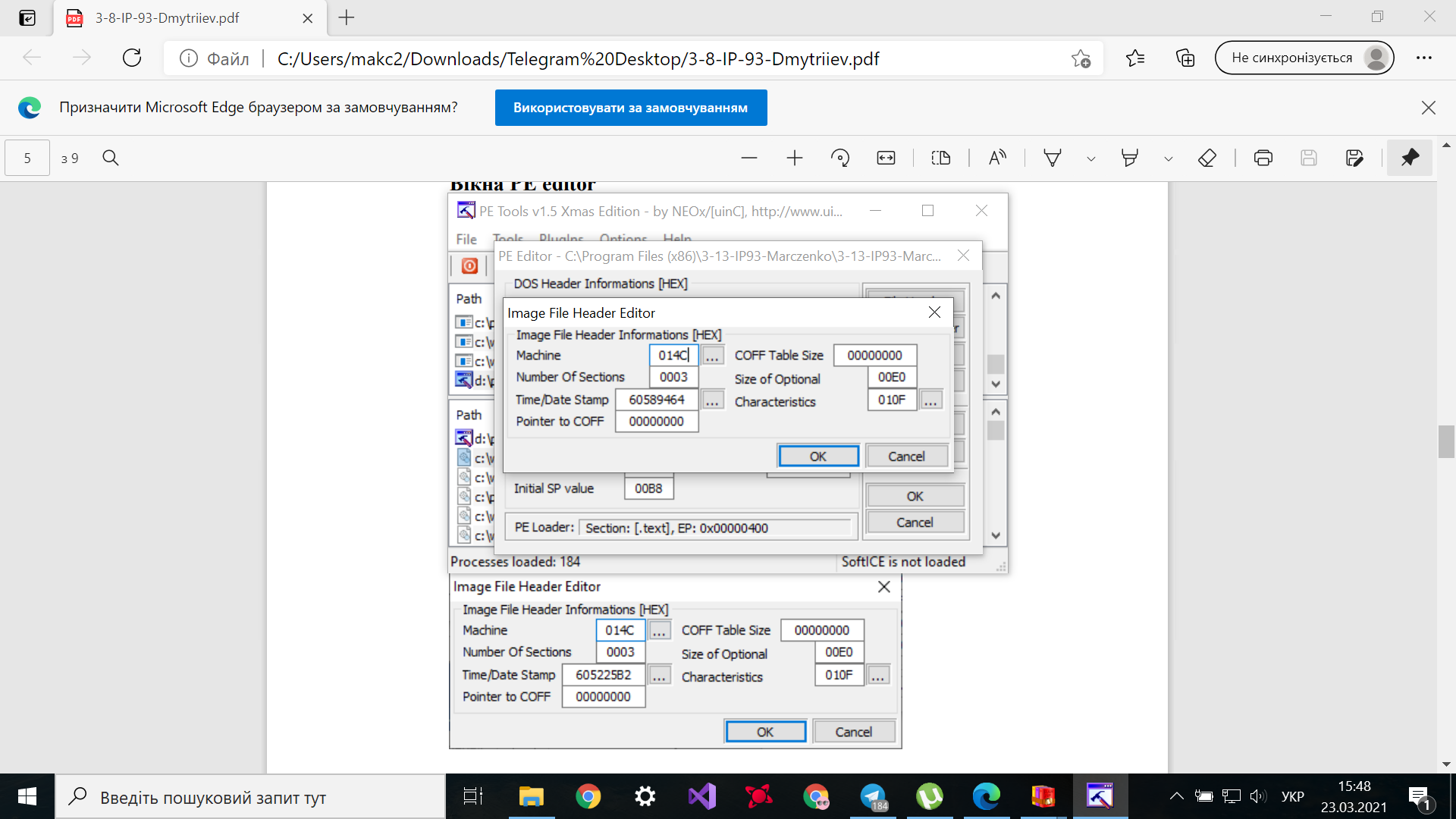
У дослідженні в шістнадцятковому форматі виділені місцеположення заголовку Dos, заголовку PE та заголовків секцій.

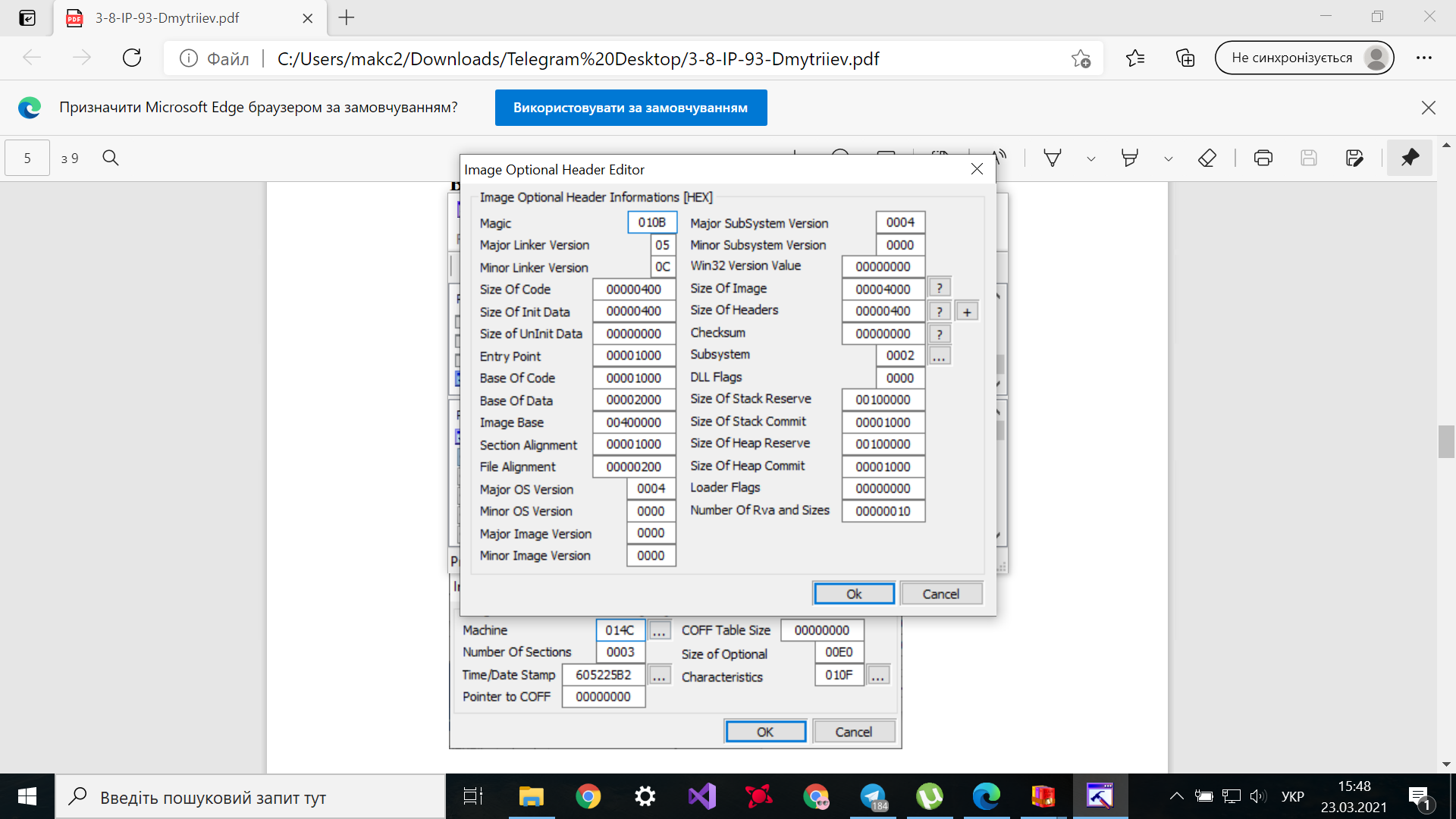
Таблиця параметрів файлу

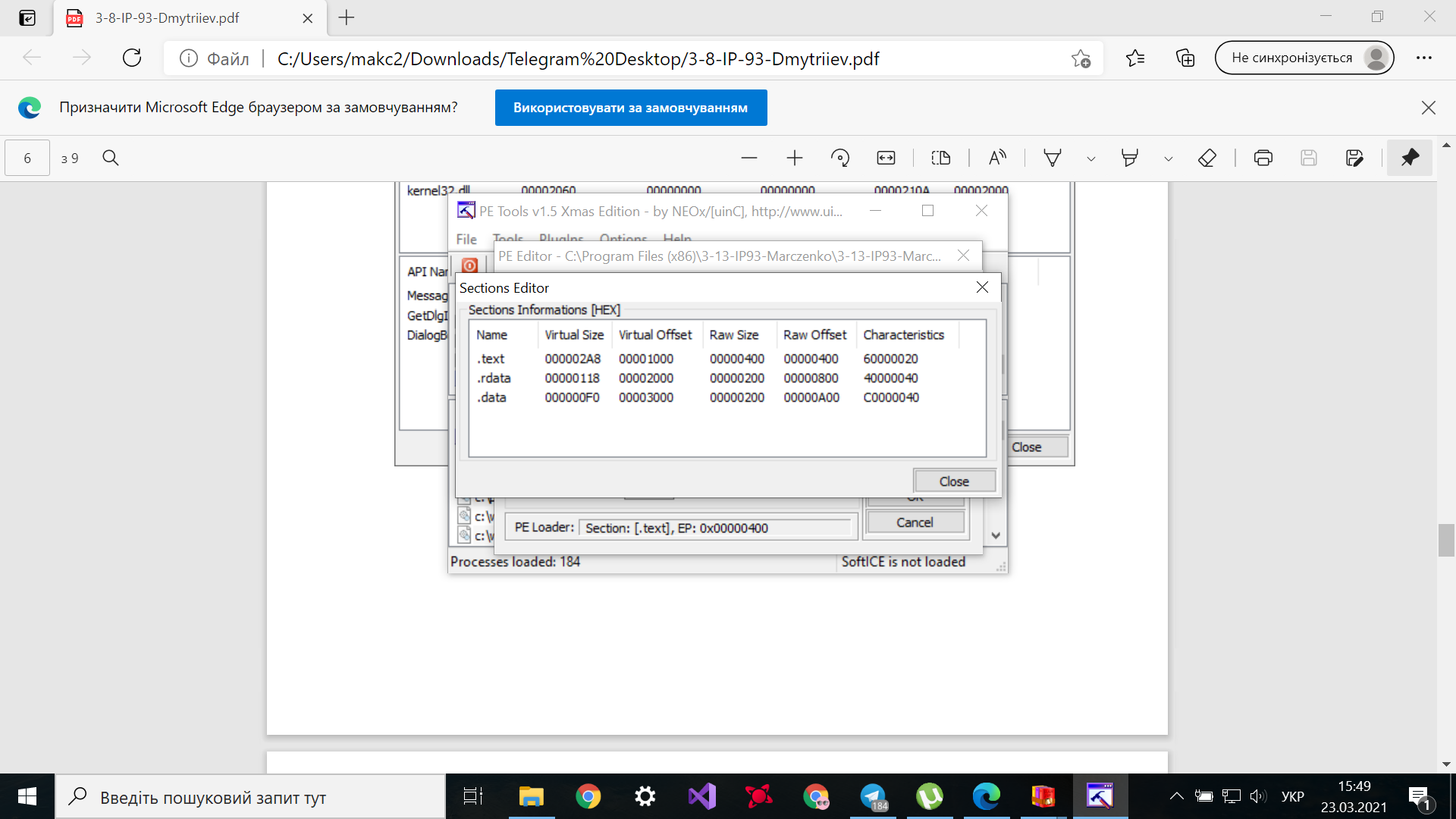
|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Значення | Опис поля | Опис значення поля |
| Machine type | 014С | Тип машини (CPU) | Intel 386 або пізніша, та інші схожі процесори |
| Optional Header standard fields | | | |
| Magic | 010В | Число типу unsigned integer, що ідентифікує стан файлу | Ідентифікатор нормального файлу типу .ехе |
| Major linker version | 05 | Номер версії основного лінкера | |
| Minor linker version | 0С | Номер версії меншого лінкера | |
| SizeOfCode | 0000-0400 | Розмір секції коду | |
| SizeOfInitializedData | 0000-0400 | Розмір секції ініціалізованих змінних | |
| SizeOfInitializedData | 0000-0000 | Розмір секції неініціалізованих змінних | |
| AddressOfEntryPoint | 0000-1000 | Адреса вхідної точки в програму, початкова адреса програми | |
| BaseOfCode | 0000-1000 | Адреса початку секції коду, коли завантажено в пам’ять | |
| BaseOfData | 0000-2000 | Адреса початку секції змінних, коли завантажено в пам’ять | |
| Section Table (Section Headers) | | | |
| Name | text | Назва секції - 8 бітний стрінг ASCII | |
| VirtualSize | 0000-02A8 | Розмір секції, завантаженої в пам’ять | |
| VirtualAdress | 0000-1000 | Адреса першого байту секції | |
| SizeOfRaw  Data | 0000-0400 | Розмір секції | |
| PointerToRaw  Data |  | Пойнтер до першої сторінки секції у файлі COFF | |
| PointerTo  Relocations | 0 | Пойнтер до початку входжень релокацій в секцію (для файлів .ехе рівний 0) | |
| PointerTo  Linenumbers |  | Пойнтер до початку входжень в секцію чисел | |
| NumberOf  Relocations | 0 | Кількість релокаційних входжень в секцію (для файлів .ехе рівний 0) | |
| NumberO  fLinenumbers |  | Кількість числових входжень у секцію | |
| Characteristics | 6000-0020 | Прапори опису характеристики секції | |

Дослідження в PE Tools





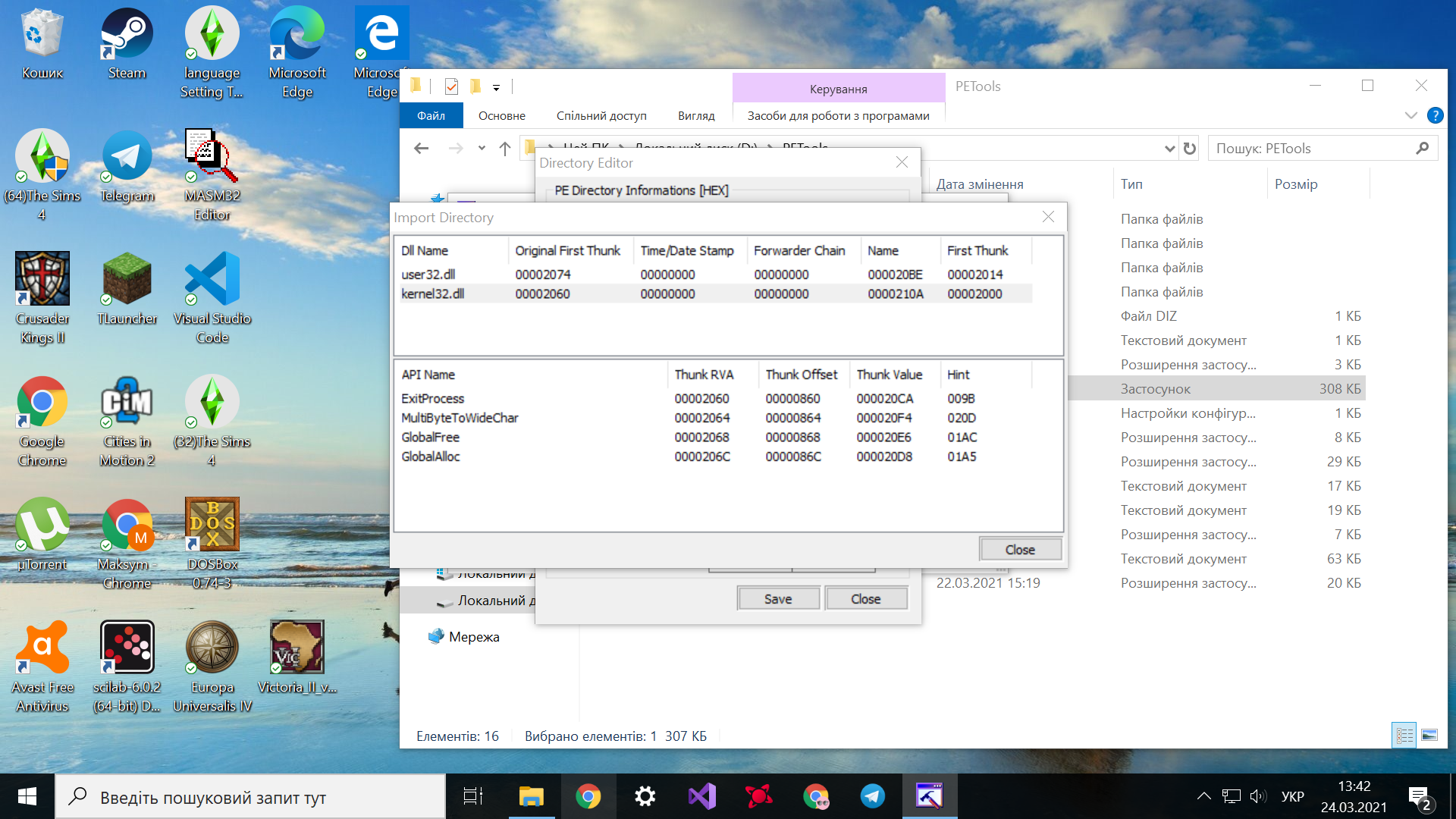




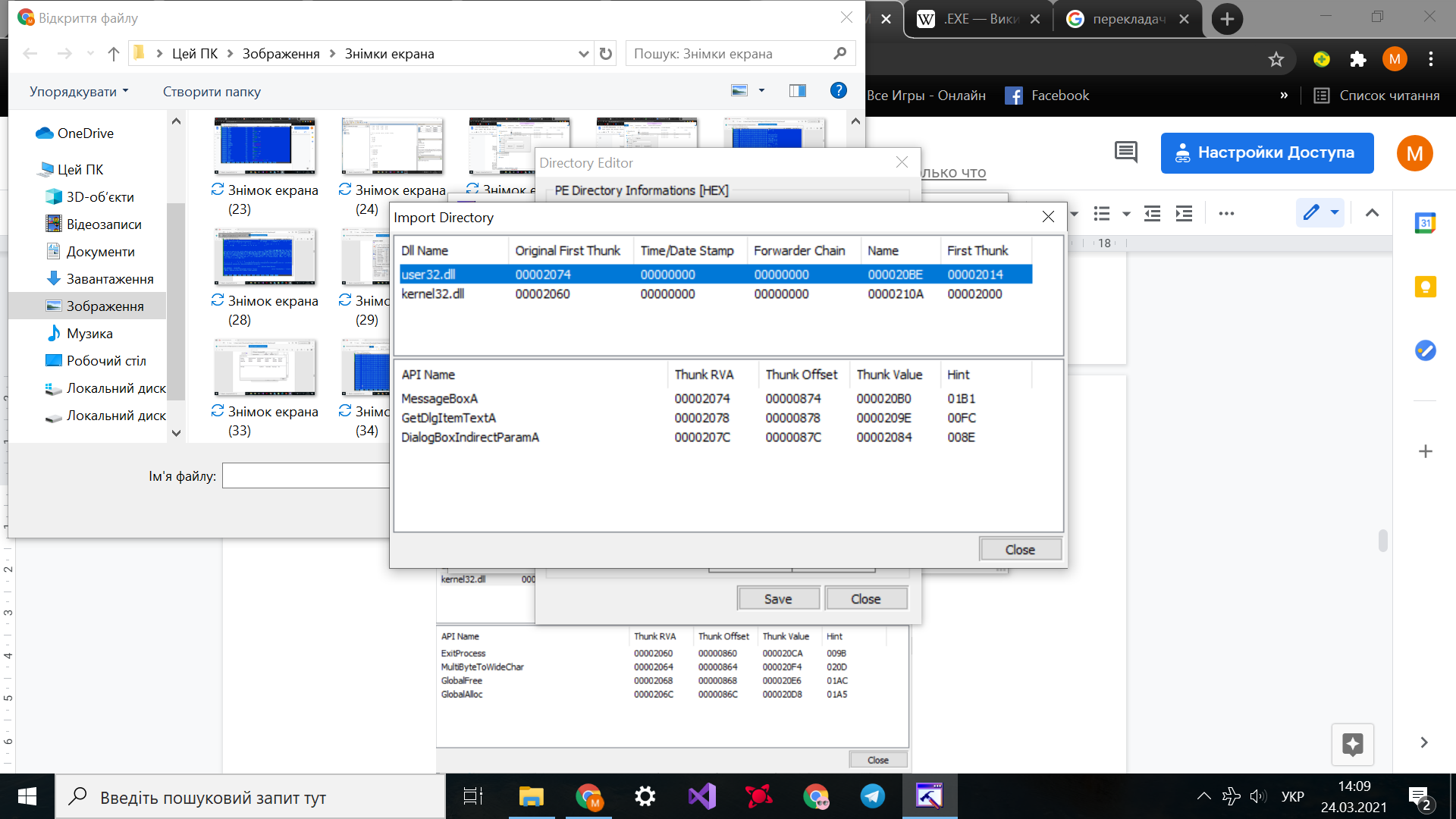
Дослідження таблиці імпорту

В програмі імпортується дві бібліотеки: user32.dll та kernel32.dll, нижче зображено список функцій цих бібліотек, що використовується в програмі, знайдений за допомогою PE Tools.

kernel32.dll

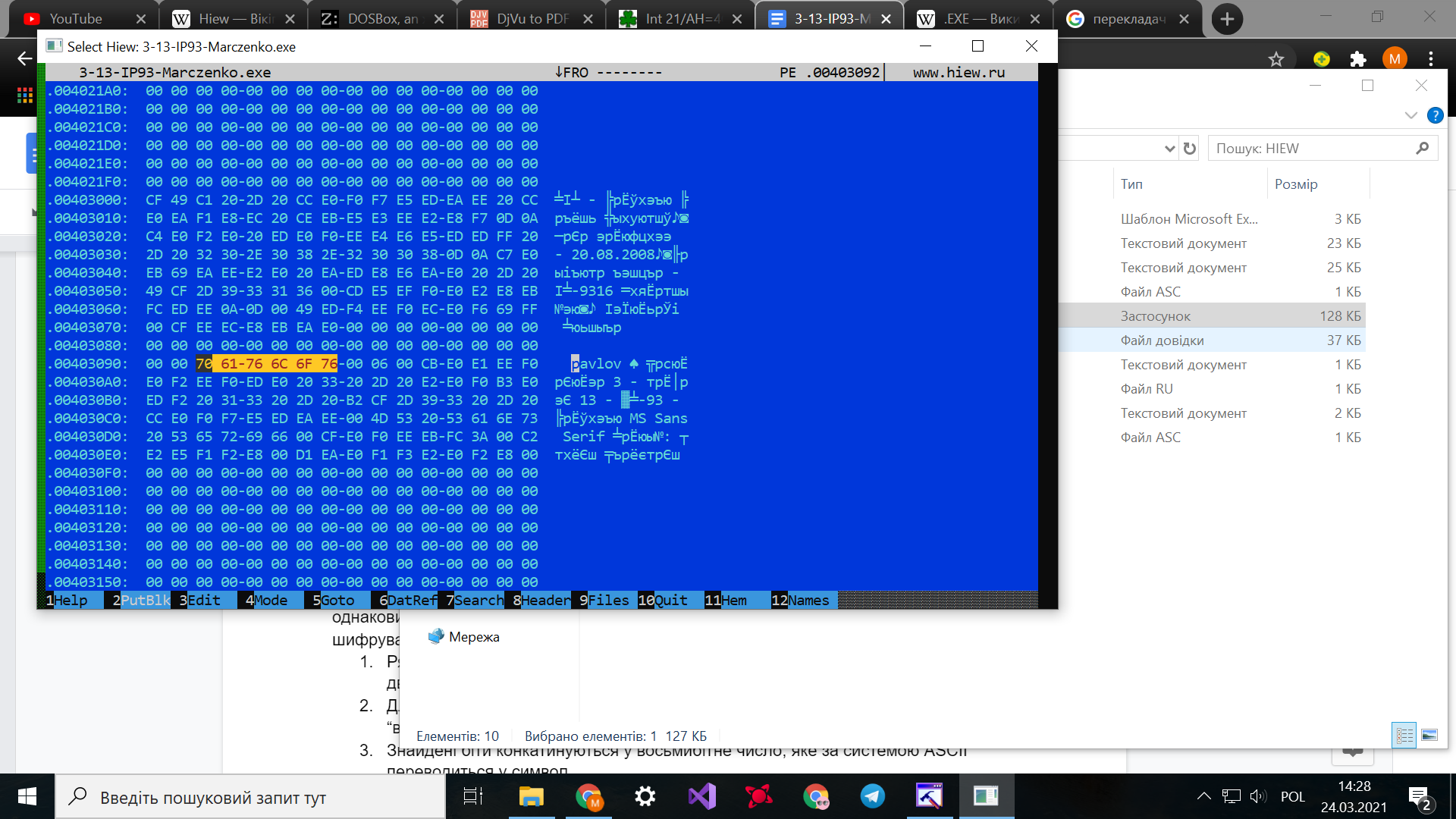


user32.dll



Виявлення паролю у тексті файлу

Щоб знайти початок секції data, де і зберігається пароль, було додано ImageBase та офсет секції data, що дало 0040-3000. Як помітно, пароль явно зберігається у файлі.



Дослідження файлу, зашифрованого методом XOR

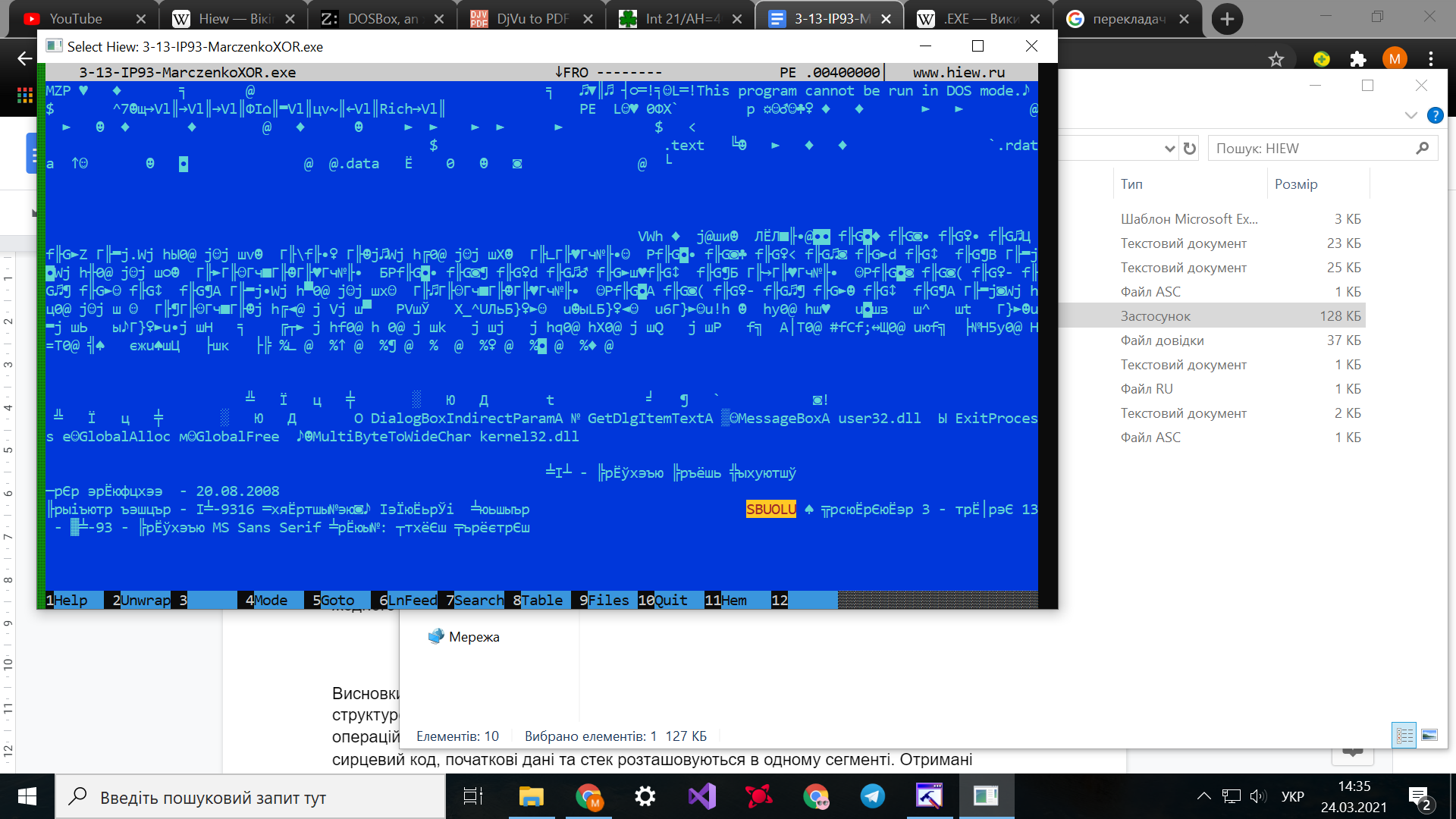
Незважаючи на те, що пароль, що потрібно ввести користувачеві у обох випадках був однаковий, у другій версії програми його не записано явно, оскільки застосовувалося шифрування методою ХОR. Шифрування відбувається таким способом:

1. Рядок розбивається на окремі літери, котрі через систему ASCII переводяться у двійкове число
2. Для кожного біту цього восьмизначного числа, застосовуємо бінарну операцію “виключне або” з певним числом (у моєму випадку це число 23h = 0010 0011).
3. Знайдені біти конкатинуються у восьмибітне число, яке за системою ASCII переводиться у символ.
4. Всі знайдені символи конкатинуються в один рядок, який і зберігається у програмі (рядок SBUOLU).

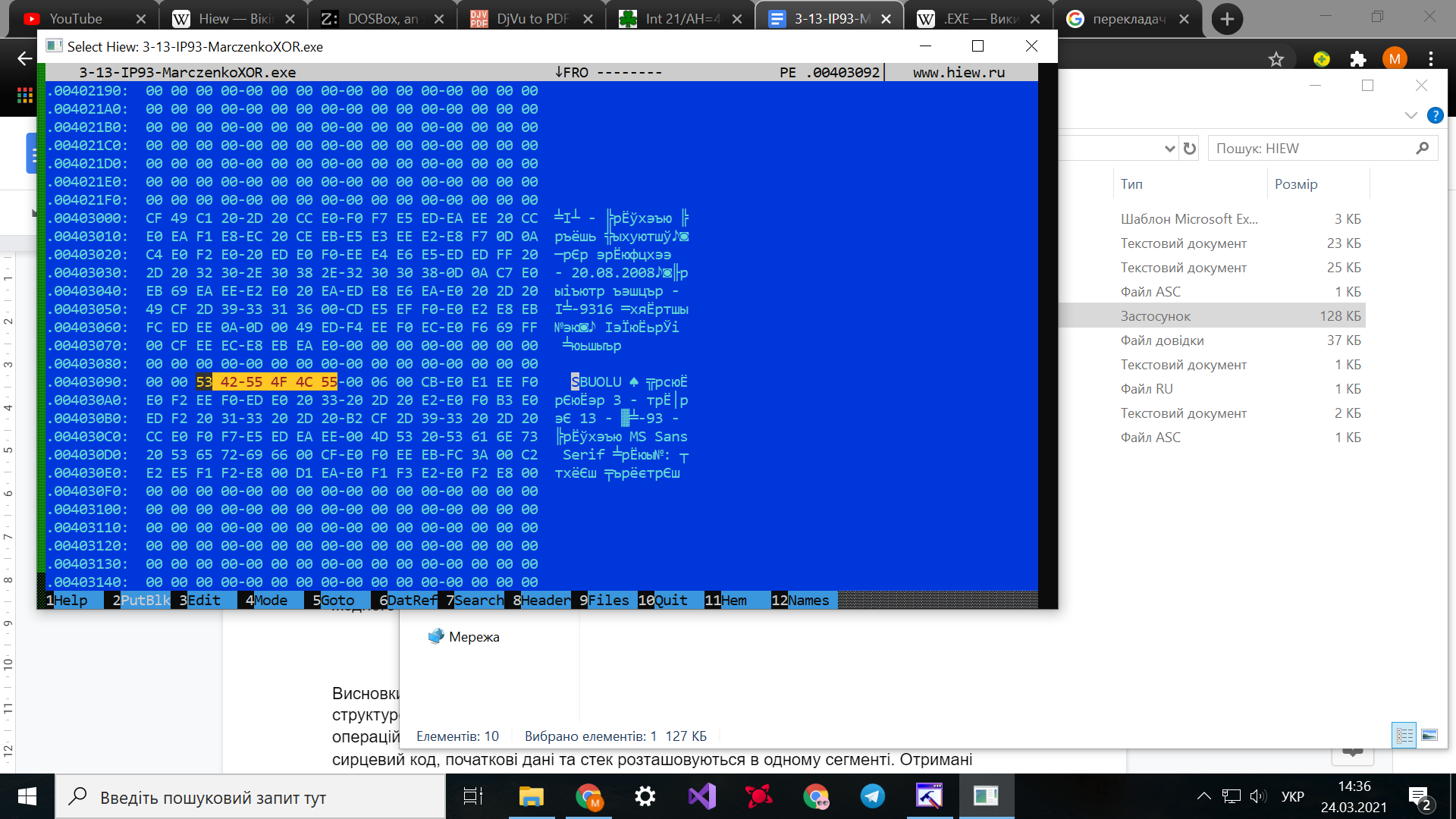
Для дешифрування повторюємо той самий алгоритм, на вході подаємо вже зашифровану версію рядка.

Даний спосіб шифрування є одним з найпростіших як для імплементації, так і для дешифрування чи взлому, проте він захищає дані краще, ніж взагалі не застосовувати жодного шифрування. На скриншотах тексту файлу в режимі text та hex виділено місце зберігання зашифрованого паролю.

Режим text



Режим hex



Висновки

Під час виконання лабораторної роботи я ознайомився з структурою файлів формату .exe - розширення виконуваного файлу, що широко застосовується в системах DOS та Microsoft Windows. Файл EXE складається з двох частин: заголовок (інформація до завантажувача програми) та модуль завантаження. Також було досліджено деякі параметри файлу, та способи їх знаходження та зміни, зокрема за допомогою програми PE Tools. За допомогою цього ж застосунку було досліджено імпортовані бібліотеки та функції.

Додаток: Код програми 3-13-IP93-MarczenkoXOR

.386

.model flat, stdcall

option casemap :none

include D:\masm32\include\windows.inc

include D:\masm32\include\dialogs.inc

include D:\masm32\macros\macros.asm

include D:\masm32\include\user32.inc

include D:\masm32\include\kernel32.inc

include D:\masm32\include\masm32.inc

includelib D:\masm32\lib\user32.lib

includelib D:\masm32\lib\kernel32.lib

includelib D:\masm32\lib\masm32.lib

DialogProcess PROTO :DWORD,:DWORD,:DWORD,:DWORD

.data

information db "ПIБ - Марченко Максим Олегович", 13, 10,

"Дата народження - 20.08.2008", 13, 10,

"Залiкова книжка - IП-9316",0

wrong\_pass db "Неправильно",10,13,0

info\_title db "Iнформацiя",0

error\_title db "Помилка",0

try\_pswd db 32 dup (0)

pswd\_sys db "SBUOLU",0

key EQU 23h

pswd\_length dw 6

.code

start:

;; initialise starting dialog

Dialog "Лабораторна 3 - варіант 13 - ІП-93 - Марченко", "MS Sans Serif",12, \

WS\_OVERLAPPED or WS\_SYSMENU or DS\_CENTER, \

4,7,7,150,90,1024

DlgStatic "Пароль:",SS\_CENTER,7,5,60,10,100

DlgEdit WS\_BORDER,7,20,100,11,1000

DlgButton "Ввести", WS\_TABSTOP,10,40,45,20,IDOK

DlgButton "Скасувати", WS\_TABSTOP,65,40,45,20,IDCANCEL

CallModalDialog 0,0,DialogProcess,NULL ;; calling starting dialog

;; configuring user's response

DialogProcess proc hWindow:DWORD,userMessage:DWORD,wParam:DWORD,lParam:DWORD

.IF userMessage == WM\_INITDIALOG

.ELSEIF userMessage == WM\_COMMAND

.IF wParam == IDOK ;; user entered password

invoke GetDlgItemText, hWindow, 1000, addr try\_pswd,512

call Compare

.ENDIF

.IF wParam == IDCANCEL ;; user cancelled entering

invoke ExitProcess,NULL

.ENDIF

.ELSEIF userMessage == WM\_CLOSE ;; user exited from the program

invoke ExitProcess,NULL

.ENDIF

return 0

DialogProcess endp

Compare proc

cld

mov bx, 0

beg\_xor: ;; cycle to decifer the password via XOR-function

CMP bx, pswd\_length

JZ cont

L1:

xor pswd\_sys[bx], key

inc bx

JMP beg\_xor

cont: ;; end of the XORcycle

mov bx, 0

beg\_check: ;; cycle to compare input with password

CMP bx, pswd\_length

JZ equal

L2:

mov al, try\_pswd[bx]

mov ah, pswd\_sys[bx] ;; comparing passwords

cmp ah , al

jne not\_equal

inc bx

equal:

invoke MessageBox, 0, addr information, addr info\_title, 0

invoke ExitProcess,NULL

ret

not\_equal:

invoke MessageBox, 0, addr wrong\_pass, addr error\_title, 0

invoke ExitProcess,NULL

ret

Compare endp

end start