**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  
«Технології розроблення системних програм»

на тему  
«Дослідження структури програм формату EXE.»

Виконав: Перевірив:

студент групи ІП-84 доц. кафедри ОТ  
Голубов Іван Олегович Павлов В. Г.  
номер залікової книжки: 8404

Київ 2020

**Мета роботи**

Ознайомлення із специфікацією COFF (Common Object File Format). Вивчення прийомів дослідження структури файлів PE-формату.

**Порядок виконання роботи**

1. Вивчити структуру програм формату **EXE** [1].

2. Розробити програму на мові Асемблер, за допомогою якої у віконному інтерфейсі по введеному паролю виводяться персональні дані студента – ПІБ, дата народження, номер залікової книжки тощо (див. лаб. роботу 1).

3. Виконати компіляцію розробленого файлу у формат **EXE**.

4. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.

5. Отриманий виконавчий файл дослідити за допомогою програми **HEX**-редактора **HIEW32** (див. лаб. роботу 2) або **WinHex** (http://rainbowsky.ru/system/winhex/ - trial версия\*) [2]

6. На скріншоте перших 25 рядків вмісту файлу обвести кольоровим олівцем або фломастером області MS-DOS заголовка (**DOS\_HEADER**), PE заголовка (**PE\_HEADER**) і таблиці секцій (**SECTION\_HEADERS**). Скріншот привести в звіті по лабораторній роботі.

7. Відповідно до опису секцій [1] скласти таблицю, в яку занести параметри свого файлу, вказані в розділах 3.3.1, 3.4.1 і 4 (перша таблиця).

8. У останньому стовпчику таблиці розшифрувати виписані значення полів заголовка файлу. Таблицю привести в звіті по лабораторній роботі.

9. Провести дослідження того ж файлу за допомогою меню "**PE Editor**" безкоштовної програми **PE Tools** (http://soft.mydiv.net/win/download-PE-Tools.html\*). Все скріншоти вікон програми з даними, відповідними раніше побудованій таблиці, привести в звіті по лабораторній роботі.

10. Дослідити таблицю імпорту (**Import Directory**) даного файлу і визначити, які саме функції використовуються з бібліотек, що підключаються. Скріншоти вікон **Import Directory** з функціями, що імпортуються, з кожного бібліотечного файлу привести в звіті по лабораторній роботі.

11. Знайти в тексті файлу по зсуву, узятому з побудованої таблиці, секцію з даними і переконатися, що текст оригінала пароля, що міститься в тексті програми, може бути легко виявлений за допомогою **HEX**-редактора. Привести скріншот цього фрагмента програми у вигляді **HEX** - коду в звіті по лабораторній роботі.

12. Виконати шифрування пароля за допомогою функції **XOR**, знову скомпілювати **EXE** -файл і переконатися, що тепер вони не виявляються явним чином в тексті виконуваного **EXE** -файлу. Привести скріншоти цієї програми в режимах «**Hex**» і «**Text**» в звіті по лабораторній роботі.

13. Зробити висновки по лабораторній роботі.

**Bat-file**

echo on

cd c:\masm32

set /P var=""

dir /s /b %var% > name.tmp

set /p mydir="" < name.tmp

cd c:\masm32\labs

ml /c /coff /Fl "%mydir%"

link /subsystem:console "3-03-IP84-Golubov.obj"

**Вихідний код**

.686

include \masm32\include\masm32rt.inc

includelib \masm32\lib\user32.lib

.data

;///////////////////////////////////////////////////////////////////////////

; Data /

;///////////////////////////////////////////////////////////////////////////

password db 03Ah, 036h, 032h, 036h ;8404

PassLen=$-password

key equ 02h

info db "Golubov Ivan,", 10,

"12.05.2001", 10,

"8404", 0

buffer db 128 dup (0)

hConsoleInput dd 0

bufferLen db 0

message0 db "Enter the password:",0

message1 db "Correct password",10, 13, 0

message2 db "Incorrect password", 0

caption db "Input", 0

.code

start:

call decrypt

;///////////////////////////////////////////////////////////////////////////

; Console /

;///////////////////////////////////////////////////////////////////////////

call AllocConsole

invoke SetConsoleTitle, offset caption

invoke GetStdHandle, STD\_INPUT\_HANDLE

mov hConsoleInput, eax

invoke ReadConsole, hConsoleInput,offset buffer, 128, offset bufferLen, NULL

;///////////////////////////////////////////////////////////////////////////

; compare password /

;///////////////////////////////////////////////////////////////////////////

lea ebx, buffer

call stringlength

mov edi, esi

mov edi, PassLen

sub edi, esi

cmp edi, 0

jne wrng

mov ecx, PassLen

lea esi, password

lea edi, buffer

repe cmpsb

je right

jmp wrng

;///////////////////////////////////////////////////////////////////////////

stringlength proc

xor esi, esi

xor edx, edx

iter:

mov eax, [ebx+esi]

inc esi

cmp eax, edx

jne iter

sub esi, 3

ret

stringlength endp

;///////////////////////////////////////////////////////////////////////////

right proc

invoke MessageBox, 0, offset info, offset message1 ,MB\_OK

invoke ExitProcess, 0

right endp

wrng proc

invoke MessageBox, 0, offset message2, offset message2 ,MB\_OK

invoke ExitProcess, 0

wrng endp

decrypt proc

xor esi, esi

hash:

xor password[esi], key

inc esi

cmp esi, PassLen

jne hash

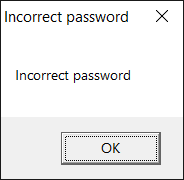
ret

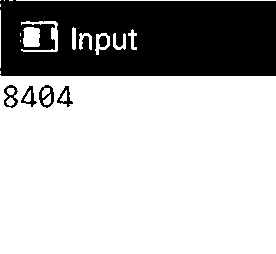
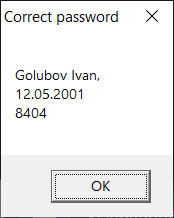
decrypt endp

;///////////////////////////////////////////////////////////////////////////

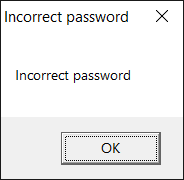
end start

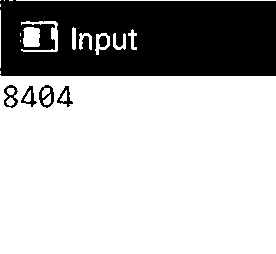
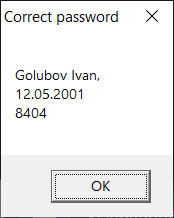
**Результати роботи програми без xor**



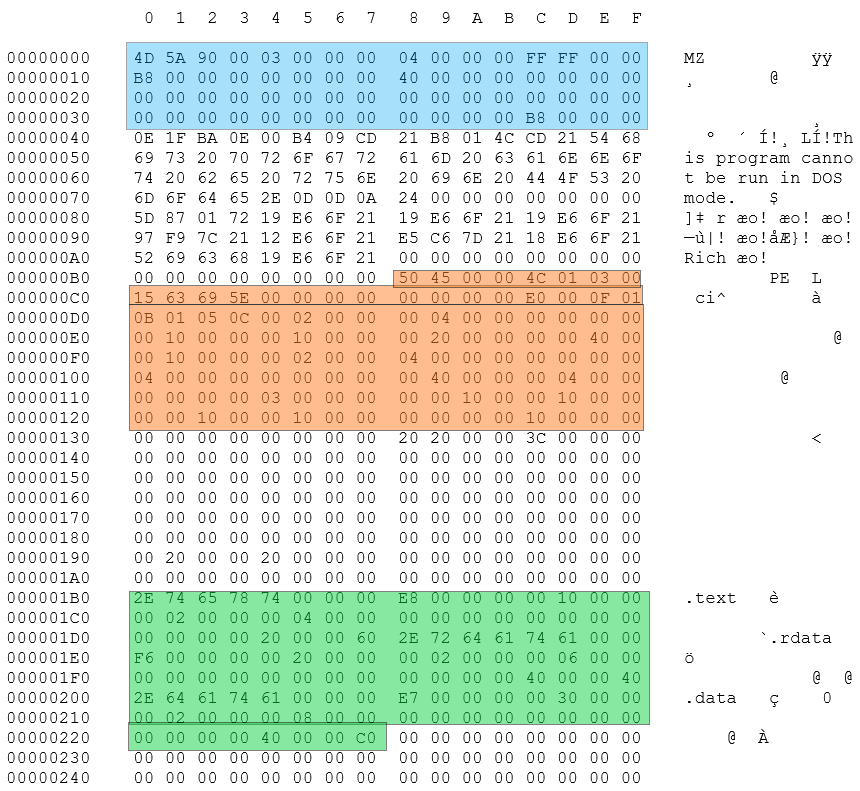
 

**Результати роботи програми з xor**



**Області розташування DOS\_HEADER PE\_HEADER SECTION\_HEADERS в лабораторній роботі**



**DOS\_HEADER PE\_HEADER SECTION\_HEADERS**

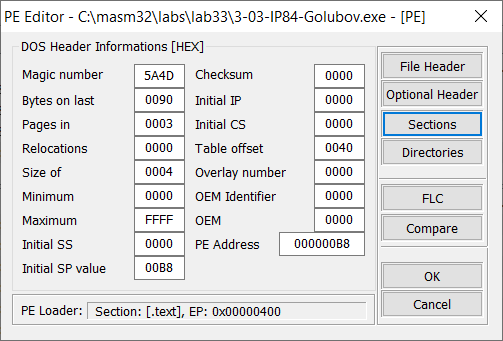
**Таблиця**

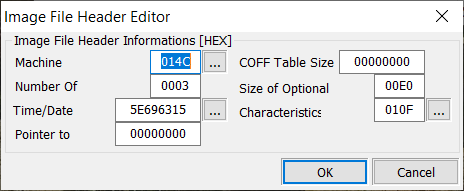
|  |  |  |  |
| --- | --- | --- | --- |
| **Поле** | **Опис поля** | **Значення поля** | **Розшифрування поля** |
| Machine | Архітектура процесора | 0x14C | Intel 386 або вище |
| Magic | Формат PE | 0x10B | PE32 |
| MajorLinkerVersion | Номер основної версії linker-а | 0x5 | 5 |
| MinorLinkerVersion | Номер допоміжної версії linker-а | 0xC | 12 |
| SizeOfCode | Сума розмірів всіх секцій, що містять програмний код | 0x200 | 512 |
| SizeOfInitializedData | Сума розмірів всіх секцій, що містять ініціалізовані дані | 0x400 | 1024 |
| SizeOfUninitializedData | Сума розмірів всіх секцій, що містять неініціалізовані дані | 0x0 | 0 |
| AddressOfEntryPoint | Адреса точки входу, щодо бази зображень, при завантаженні в пам'ять | 0x1000 | 4096 |
| BaseOfCode | Адреса, щодо бази зображень, початку розділу коду при завантаженні в пам'ять | 0x1000 | 4096 |
| BaseOfData | Адреса, щодо бази зображень, початку розділу даних при завантаженні в пам'ять | 0x2000 | 8192 |
| Name | Назва секції | 74 78 65 74 2E | .text |
| 61 74 61 64 72 2E | .rdata |
| 61 74 61 64 2E | .data |
| VirtualSize | Розмір секції в пам'яті | 0xE8 | 232 |
| 0xF6 | 246 |
| 0xE7 | 231 |
| VirtualAddress | Адреса першого байту секції | 0x1000 | 4096 |
| 0x2000 | 8192 |
| 0x3000 | 12288 |
| SizeOfRawData | Розмір ініціалізованих даних на диску | 0x200 | 512 |
| 0x200 | 512 |
| 0x200 | 512 |
| PointerToRawData | Вказівник файлу на першу сторінку розділу в файлі coff | 0x400 | 1024 |
| 0x600 | 1536 |
| 0x800 | 2048 |
| PointerToRelocations | Покажчик файлу на початок записів переміщення для розділу. | 0x0 | 0 |
| 0x0 | 0 |
| 0x0 | 0 |
| PointerToLinenumbers | Покажчик файлу на початок записів номера рядка для розділу. | 0x0 | 0 |
| 0x0 | 0 |
| 0x0 | 0 |
| NumberOfRelocations | Кількість записів про переміщення для розділу. | 0x0 | 0 |
| 0x0 | 0 |
| 0x0 | 0 |
| NumberOfLinenumbers | Кількість записів на номер рядка для розділу. | 0x0 | 0 |
| 0x0 | 0 |
| 0x0 | 0 |
| Characteristics | Прапорці, що описують вміст даної секції | 0x60000020 | Секція містить виконуваний код. Можна як виконувати її код, так і читати |
| 0x40000040 | Секція містить початкові дані, які можна читати. |
| 0xС0000040 | Секцію можна читати і в неї писати. Вона містить ініціалізовані дані. |

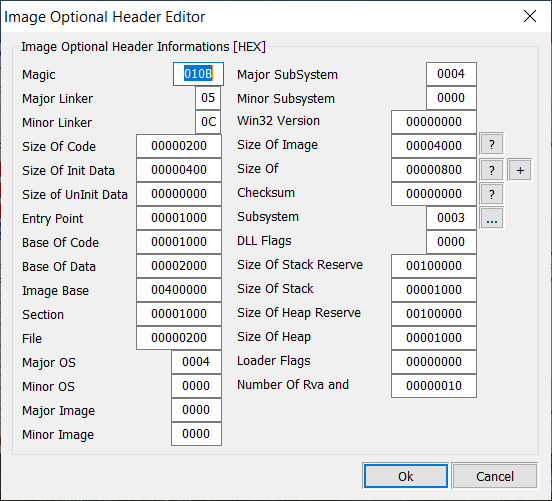
**Опис знаходження значень полів для таблиці**

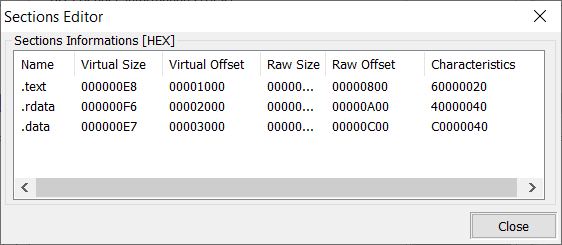
|  |  |
| --- | --- |
| Machine | 2 байти в Pe-header після 50 45 00 00 |
| Magic | Після Pe-header(розмір 20 байт) перші 2 байти в OptionalHeader |
| MajorLinkerVersion | Зміщення 2h і розмір 1 байт |
| MinorLinkerVersion | Зміщення 3h і розмір 1 байт |
| SizeOfCode | Зміщення 4h і розмір 4 байти |
| SizeOfInitializedData | Зміщення 8h і розмір 4 байти |
| SizeOfUninitializedData | Зміщення Сh і розмір 4 байти |
| AddressOfEntryPoint | Зміщення 10h і розмір 4 байти |
| BaseOfCode | Зміщення 14h і розмір 4 байти |
| BaseOfData | Зміщення 18h і розмір 4 байти |
| Name | 8 байт на початку секції , якщо є, кожні 8 байт зі зміщенням 28h |
| VirtualSize | Зміщення 8h і розмір 4 байти |
| VirtualAddress | Зміщення Сh і розмір 4 байти |
| SizeOfRawData | Зміщення 10h і розмір 4 байти |
| PointerToRawData | Зміщення 14h і розмір 4 байти |
| PointerToRelocations | Зміщення 18h і розмір 4 байти |
| PointerToLinenumbers | Зміщення 1Сh і розмір 4 байти |
| NumberOfRelocations | Зміщення 20h і розмір 2 байти |
| NumberOfLinenumbers | Зміщення 22h і розмір 2 байти |
| Characteristics | Зміщення 24h і розмір 4 байти |

**Скріншоти для підтвердження коректності таблиці**

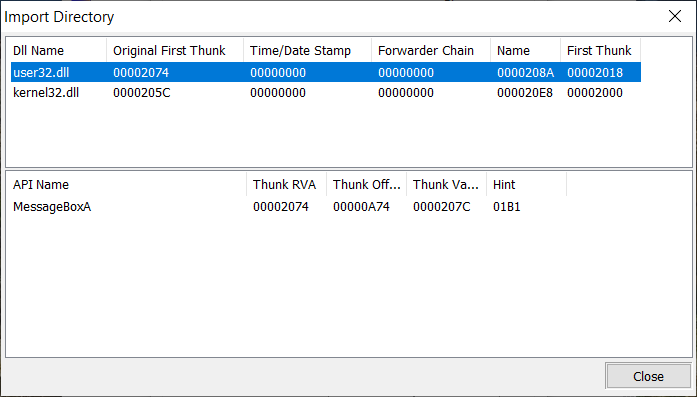


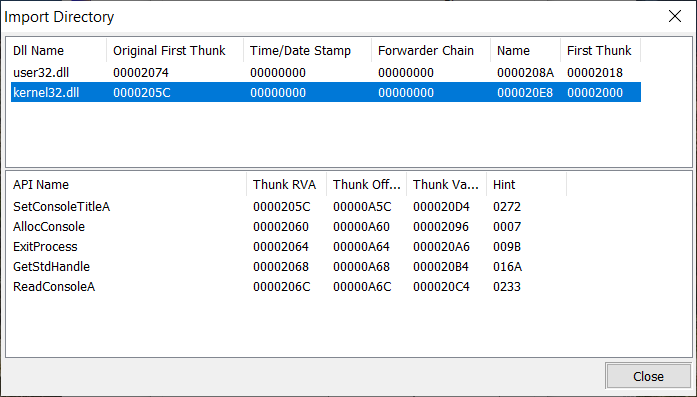




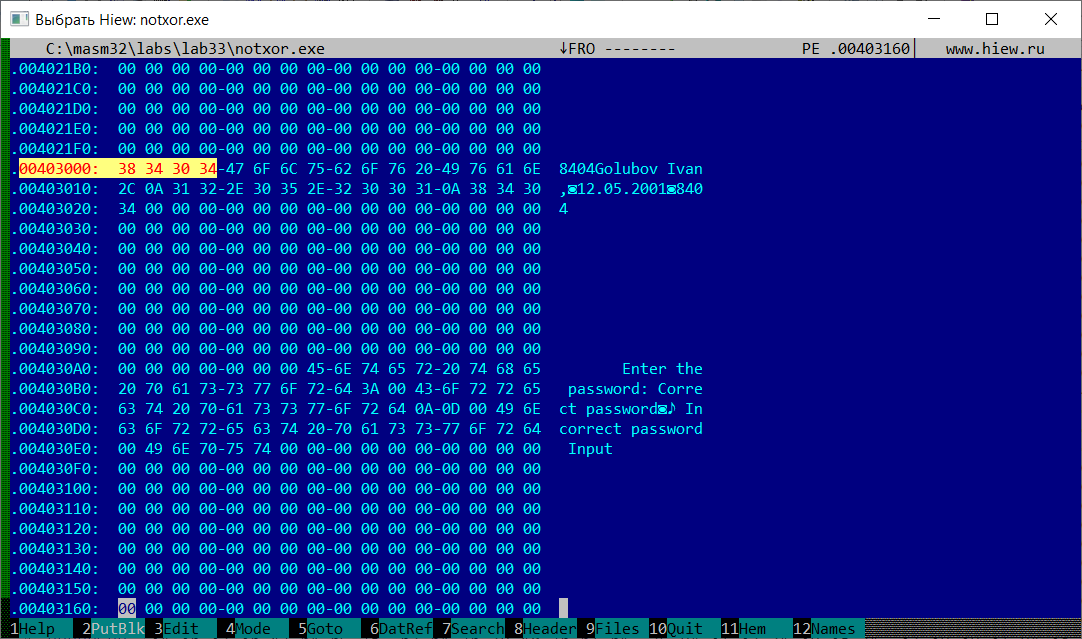


**Скріншоти вікон Import Directory з функціями, що імпортуються**



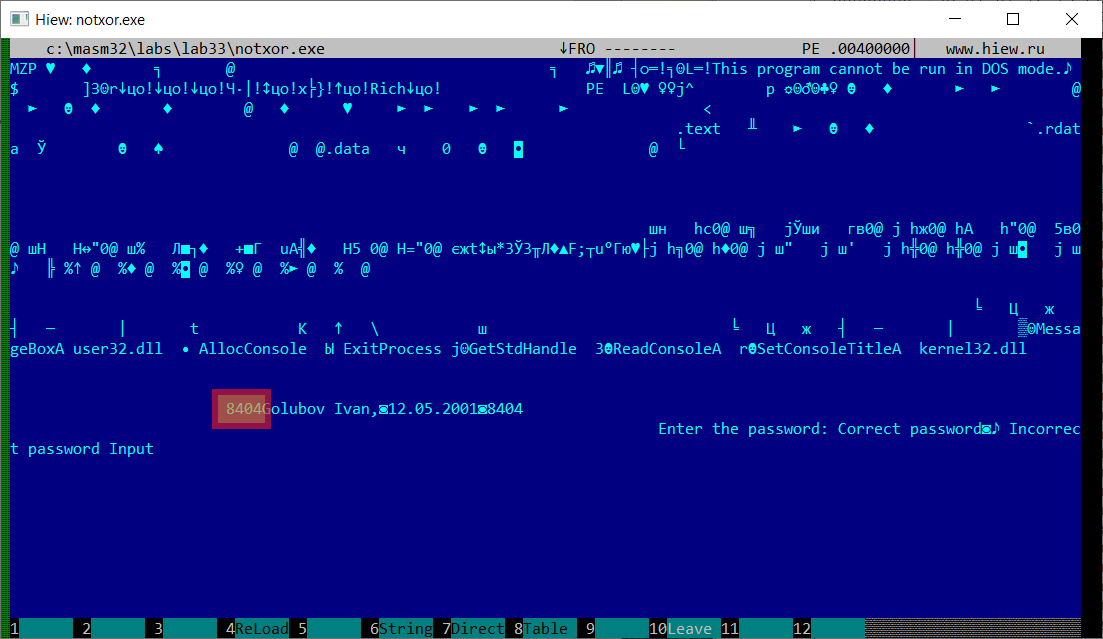


**Секція з даними, яка знайдена в тексті файлу по зсуву, узятому з побудованої таблиці**

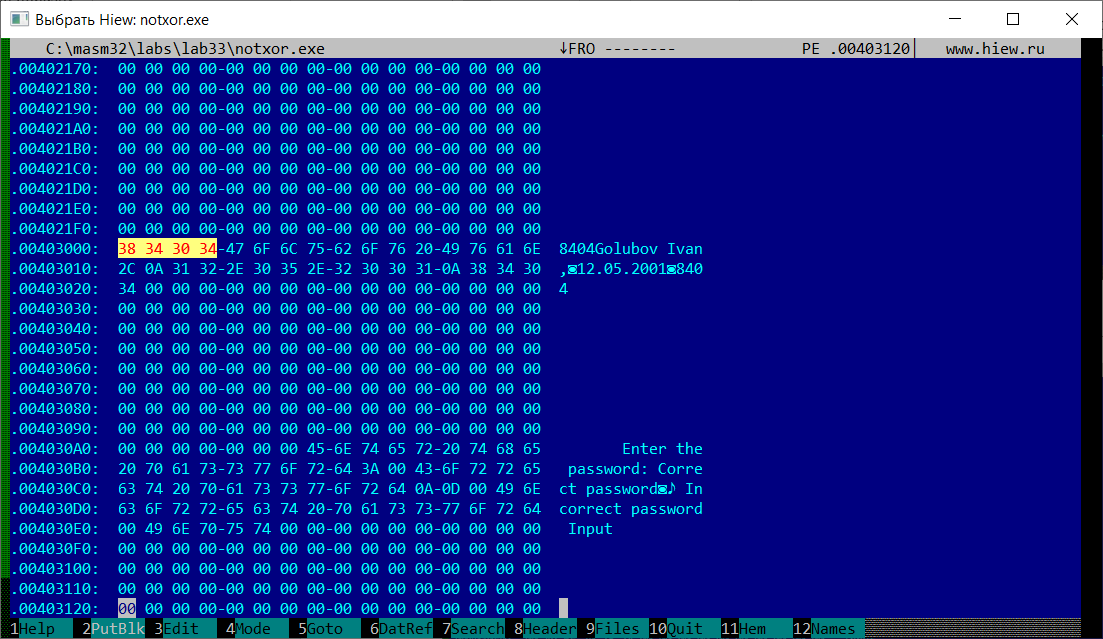


**Скріншоти без та з шифруванням**

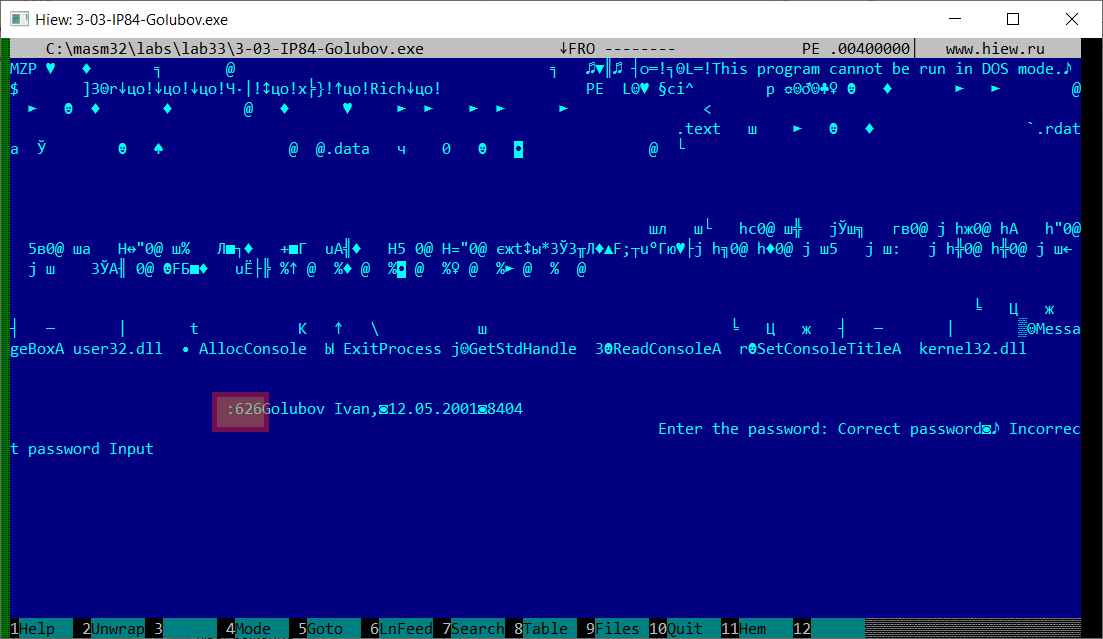
Default-Text:

****

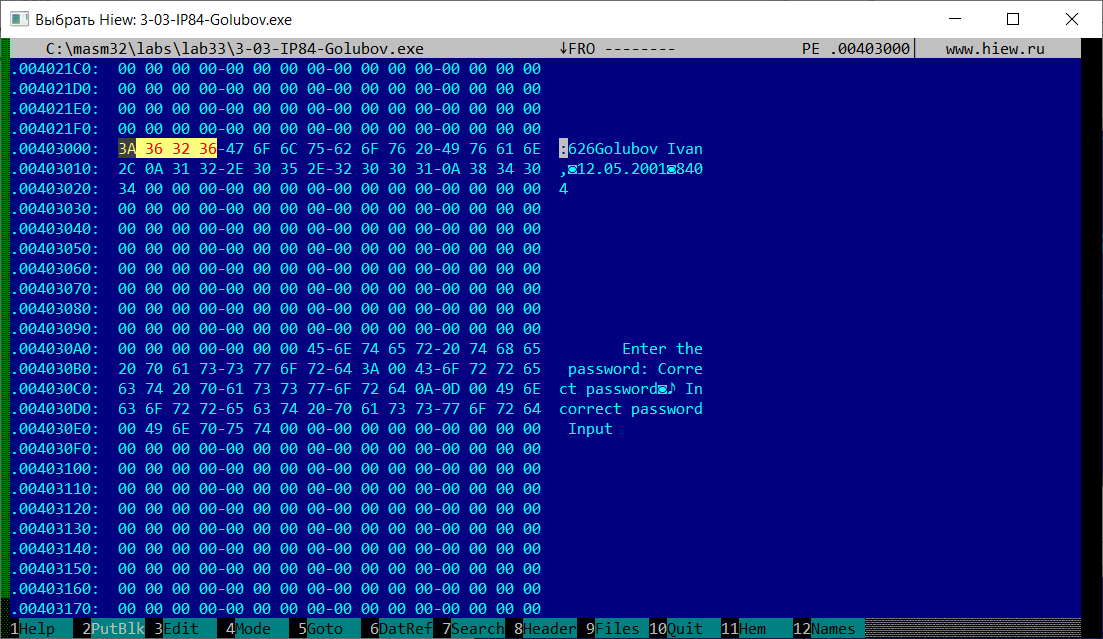
Default-Hex:



XOR-Text:

****

XOR -Hex:



**Висновки**

У ході виконання лабораторної роботи я ознайомився зі специфікацією **COFF** (Common Object File Format). Я розробив програму на мові Асемблер, за допомогою якої у віконному інтерфейсі по введеному паролю виводяться мої персональні дані. Виконав компіляцію розробленого файлу у формат EXE. Отриманий виконавчий файл я дослідив за допомогою програми HEX-редактора. Я розглянув різні заголовки, зокрема DOS - header, PE – header, Section – headers. Склав таблицю, в яку заніс і розшифрував параметри свого файлу. Провів дослідження того ж файлу за допомогою меню "PE Editor". Дослідив таблицю імпорту (Import Directory) даного файлу і визначив, які саме функції використовуються з бібліотек, що підключаються. Знайшов в тексті файлу по зсуву, узятому з побудованої таблиці, секцію з даними і переконався, що текст оригінала пароля, що міститься в тексті програми, може бути легко виявлений за допомогою HEX-редактора. Моя програма мала два варіанти: з шифруванням паролю і без. Я визначив, що при шифруванні паролю його неможливо знайти в виконуваному файлі через HEX-редактор.