Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. 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 2 курсу ФІОТ доц. Павлов В.Г.

групи ІМ-33

Родіна Олександр Вікторович

номер у списку групи - 19

Київ 2025

**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 виводяться персональні дані студента –

ПІБ, дата народження, номер залікової книжки тощо (див. лаб. ро-

боту 1), але лише при вірно введеному паролі.

3. Виконати компіляцію розробленого файлу у формат EXE.

4. Перевірити роботу програми шляхом введення як правиль-

ного, так і невірного паролів.

5. Отриманий виконавчий файл дослідити за допомогою про-

грами HEX-редактора HIEW32 (https://uk.wikipedia.org/wiki/Hiew)

або WinHex (http://rainbowsky.ru/system/winhex/ - trial версия\*)[2].

6. На скріншоте перших 25 рядків вмісту файлу обвести ко-

льоровим олівцем або фломастером області MS-DOS заголовка

(DOS\_HEADER), PE заголовка (PE\_HEADER) і таблиці секцій

(SECTION\_HEADERS). Скріншот привести в звіті по лаборатор-

ній роботі.

7. Відповідно до опису секцій [1] скласти таблицю, в яку за-

нести параметри свого файлу, вказані в розділах 3.3.1, 3.4.1 і 4 (пе-

рша таблиця).

8. У останньому стовпчику таблиці розшифрувати виписані

значення полів заголовка файлу. Таблицю привести в звіті по лабо-

раторній 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 табли-

ці, привести в звіті по лабораторній роботі.

10

10. Дослідити таблицю імпорту (Import Directory) даного

файлу і визначити, які саме функції використовуються з бібліотек,

що підключаються. Скріншоти вікон Import Directory з функція-

ми, 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 паро-

ля, 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 -

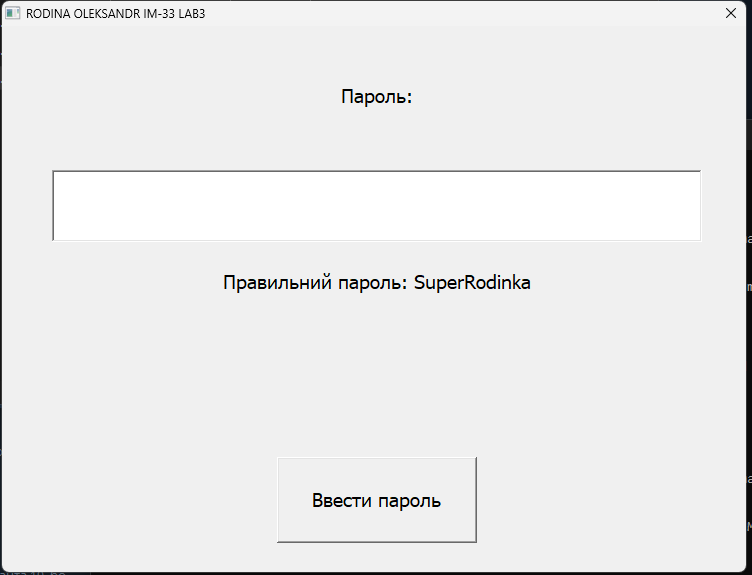
файлу. Привести скріншоти цієї програми в режимах «Hex» і

«Text» в звіті по лабораторній роботі.

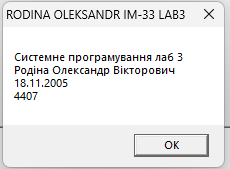
13. Зробити висновки по лабораторній роботі.

**Хід виконання роботи**

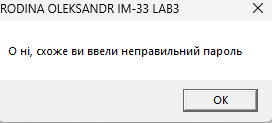
**Перевірка новоствореної програми (пункт 4)**

Вікно для паролю ****

Вікно, яке виводиться як результат правильного вводу паролю



Вікно, яке виводиться як результат неправильного вводу паролю

****

**Дослідження файлу в HIEW32 (Пункт 6)**

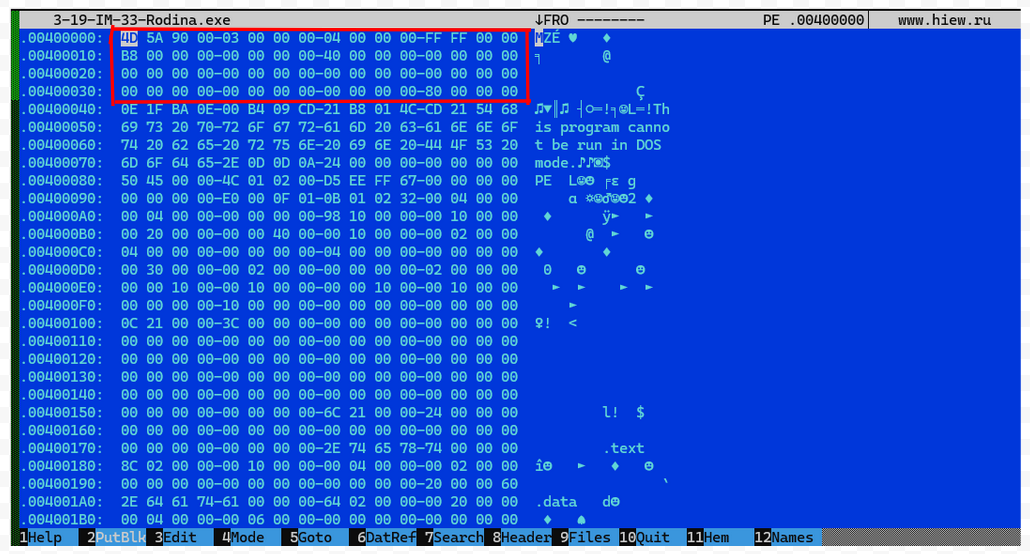
Відповідність кольорів (відповідні константи обведенні відповідним кольором):

Червоний - DOS\_HEADER

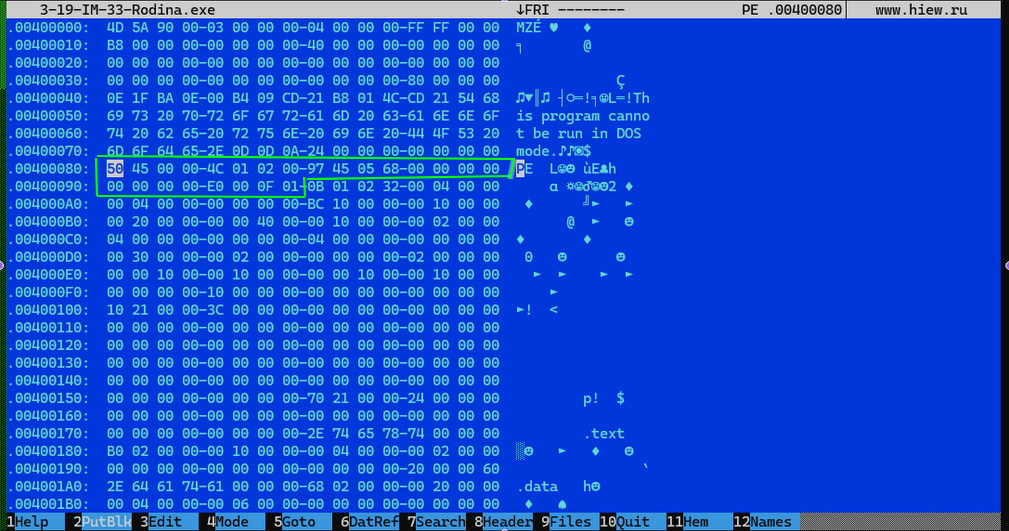
Зелений - PE\_HEADER

Рожевий - SECTION\_HEADERS

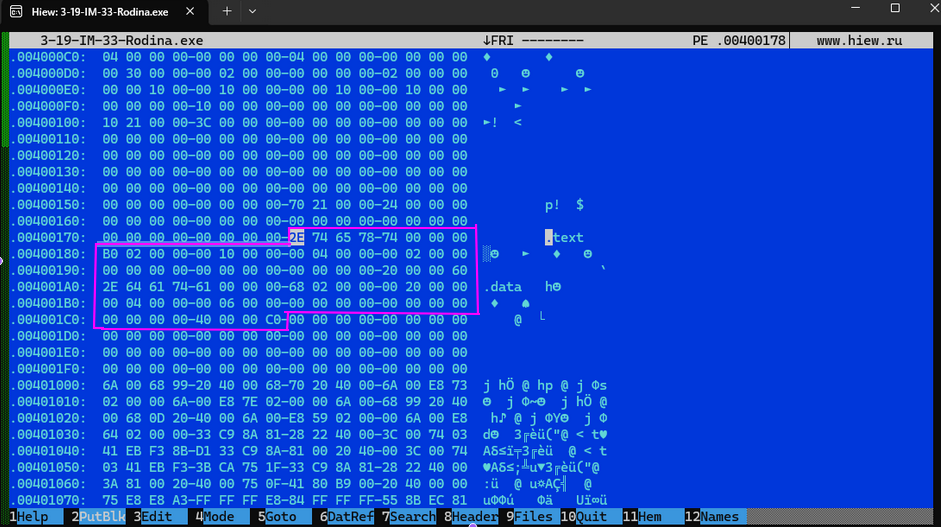
DOS\_HEADER завжди знаходиться в початку файлу та займає 64 байти



Знайдемо зсув початку PE\_HEADER за зсувом 0x3C. Цей зсув буде 80 00 00 00. Розвернемо його і отримаємо 00 00 00 08, тому зсув буде дорівнювати 128 байтів. PE\_HEADER має сталий розмір, який дорівнює 24 байт.



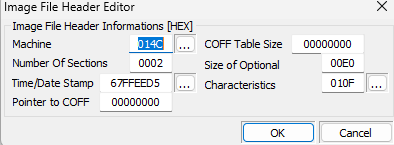
Знайдемо SizeOfOptionalHeader (E0 00). Кількість секцій можемо також знайти в PE\_HEADER. У мене це 02 00, тому і секцій 2. Розмір SECTION\_HEADERS дорівнює кількість секцій \* 40 байтів = 2 \* 40 = 80 байтів.

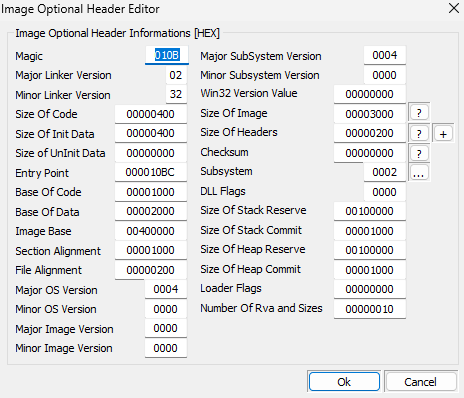
****

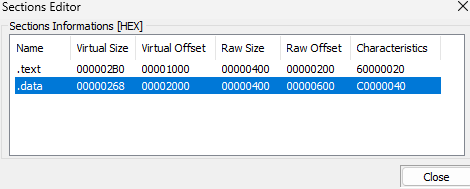
**Створення таблички (пункт 7 і пункт 8)**

| Параметр | Опис | Значення |
| --- | --- | --- |
| Machine type | Архітектура процесора, для якої призначений файл | 014C |
| Magic | Сигнатура, що ідентифікує формат файлу | 010B |
| MajorLinkerVersion | Основна версія лінкера, який створив файл | 02 |
| MinorLinkerVersion | Другорядна версія лінкера | 32 |
| SizeOfCode | Розмір виконуваного коду в файлі | 00000400 |
| SizeOfInitializedData | Розмір ініціалізованих даних | 00000400 |
| SizeOfUninitializedData | Розмір неініціалізованих даних | 00000000 |
| AddressOfEntryPoint | Адрес точки входу | 000010BC |
| BaseOfCode | Базова адреса секції коду в пам’яті | 00001000 |
| BaseOfData | Базова адреса секції даних | 00002000 |
| Name | Назва секції | .data |
| VirtualSize | Розмір секції в пам’яті під час виконання | 00000268 |
| VirtualAddress | Віртуальна адреса початку секції в пам’яті | 00002000 |
| SizeOfRawData | Розмір секції в файлі на диску | 00000400 |
| PointerToRawData | Зміщення секції у файлі відносно початку | 00000600 |
| PointerToRelocations | Зміщення таблиці перерозміщення секції | 00000000 |
| PointerToLinenumbers | Зміщення таблиці номерів рядків | 00000000 |
| NumberOfRelocations | Кількість записів перерозміщення у секції | 0000 |
| NumberOfLinenumbers | Кількість записів номерів рядків | 0000 |
| Characteristics | Атрибути секції | C0000040 |

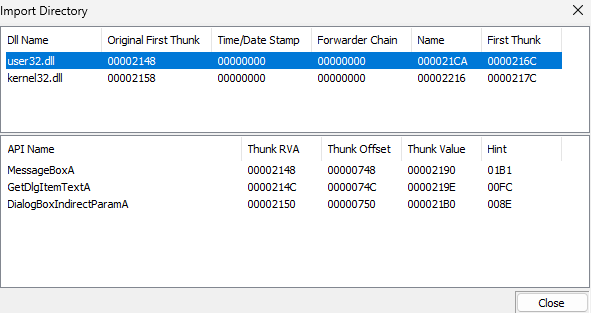
**Дослідження за допомогою PE Editor (пункт 9-10)**

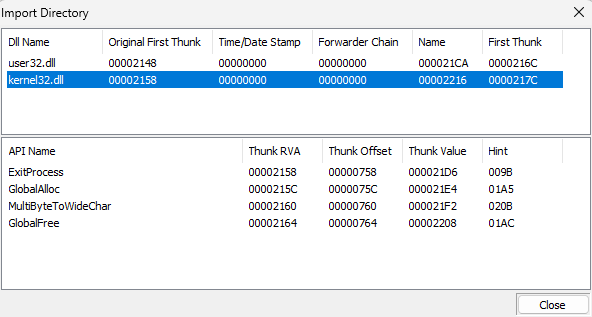
****

****

****

**import-директорії**

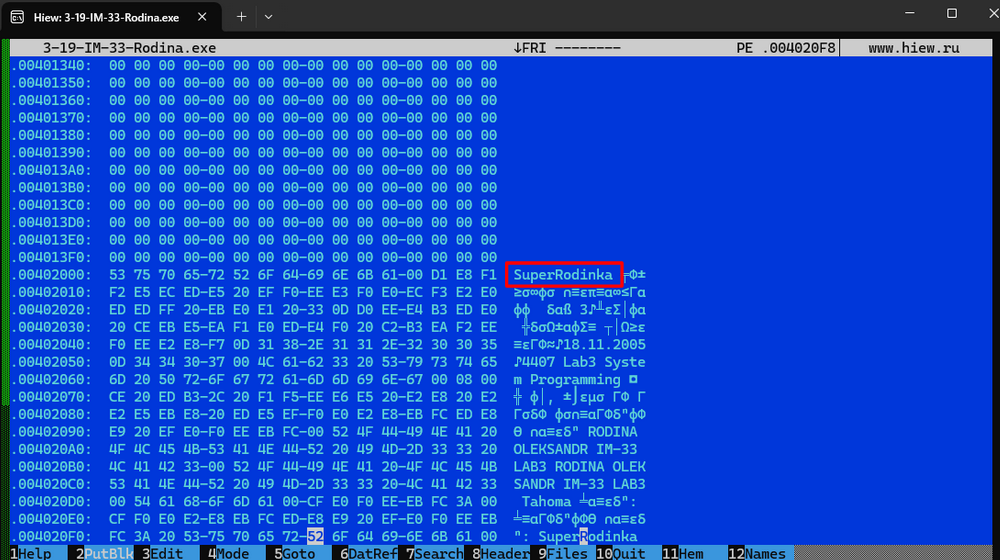
****

****

**Знайдемо пароль в HEX Editor (Пункт 11)**

Відповідність кольорів:

Червоний - Наш пароль (SuperRodinka)

  
  
Ми можемо бачити, що без шифрування будь-який зловмисник може знайти наш пароль, тому шифрування необхідне.

**Шифрування за допомогою функції XOR (Пункт 12)**

Початковий пароль:SuperRodinka

Ключ: \*\*\*\*\*\*\*\*\*\*\*\*

Шифрування:

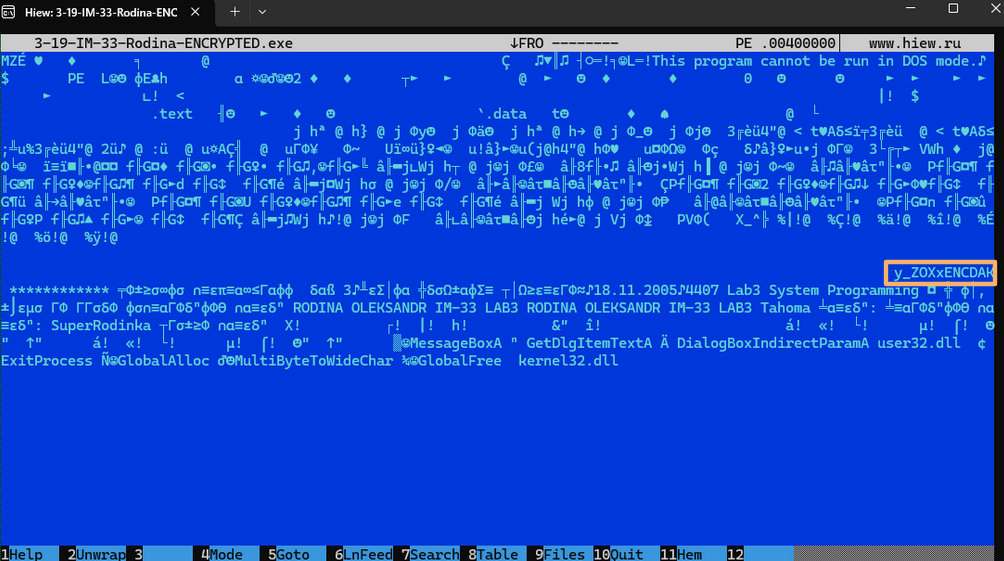
| Символ | Бінарне | Ключ | XOR результат | Новий символ |
| --- | --- | --- | --- | --- |
| S | 01010011 | 00101010 | 01111001 | y |
| u | 01110101 | 00101010 | 01011111 | \_ |
| p | 01110000 | 00101010 | 01011010 | Z |
| e | 01100101 | 00101010 | 01001111 | O |
| r | 01110010 | 00101010 | 01011000 | X |
| R | 01010010 | 00101010 | 01111000 | x |
| o | 01101111 | 00101010 | 01000101 | E |
| d | 01100100 | 00101010 | 01001110 | N |
| i | 01101001 | 00101010 | 01000011 | C |
| n | 01101110 | 00101010 | 01000100 | D |
| k | 01101011 | 00101010 | 01000001 | A |
| a | 01100001 | 00101010 | 01001011 | K |

Зашифрована версія паролю: y\_ZOXxENCDAK

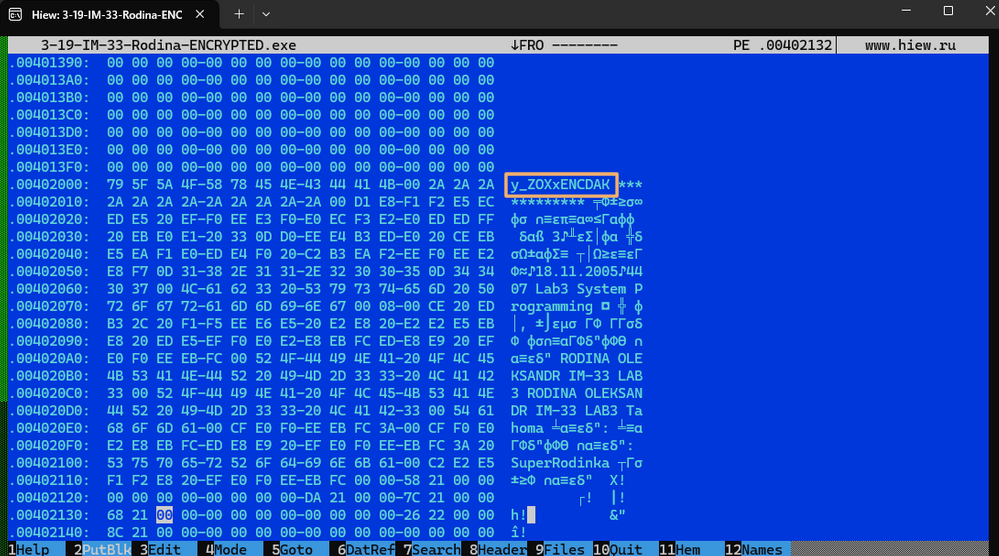
Відповідність кольорів:

Помаранчевий - Наш зашифрований пароль

**Text**



**Hex**



**Висновки**

1. Дана лабораторна робота принесла мені дуже багато нових знань по різноманітним темам, в особливості щодо дослідження exe файлів.
2. Я дізнався про шифрування паролів, зрозумів наскільки важливо використовувати шифрування для безпеки користувача та його даних. Також я навчився шифрувати паролі за допомогою функцій XOR.
3. Навчився користуватися корисними додатками HIEW32 та PE-Tools. За допомогою першого я навчився аналізувати та розуміти структуру EXE файлів, а також навіть знаходити можливі вразливості, пов’язані з відсутністю шифрування паролів. За допомогою ж PE-Tools можна дізнатися багато різноманітної інформації про структуру даного EXE файлу
4. Ознайомлення з функцією XOR призвело на мене велике враження і допомогло зрозуміти наскільки різноманітними та варіативними можуть бути метод шифрування паролю. Також я навчився шифрувати паролі за допомогою цієї функції, а також реалізовувати логіку перевірки паролів зашифрованих цією функцією.