**Лабораторна робота №3**

з дисципліни «Системне програмування» з теми

«Дослідження структури програм формату EXE.»

Виконав: Перевірив:

Студент групи ІМ-31 доц. Павлов В. Г.

Булах Богдан Валерійович

номер в списку групи: 2

Київ 2025

**Мета роботи**

Ознайомлення із специфікацією COFF (Common Object File

Format). Вивчення прийомів дослідження структури файлів PE-

формату.

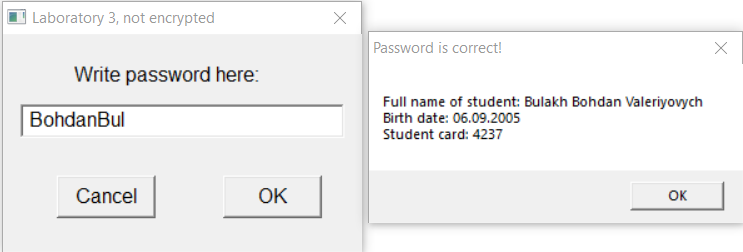
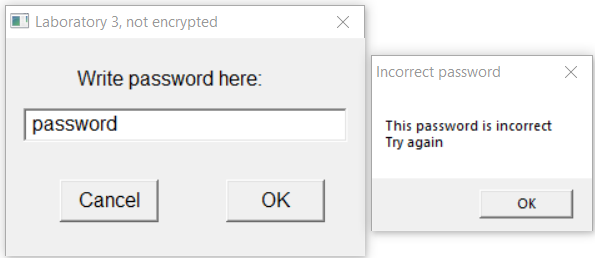
**Порядок виконання роботи:**

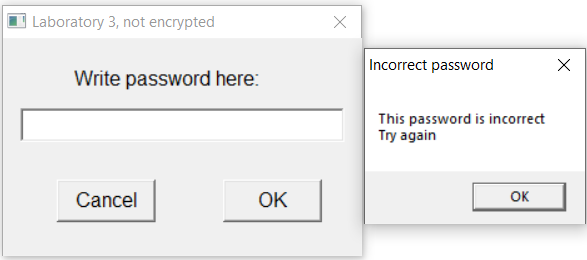
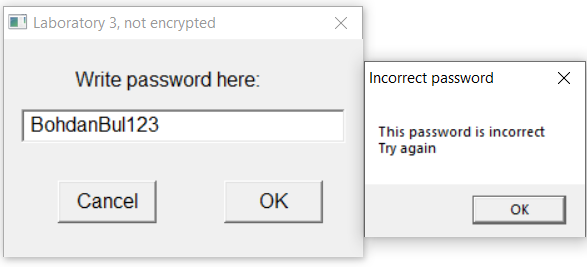
1. Вивчити структуру програм формату EXE [1]
2. Розробити програму на мові Асемблер, за допомогою якої у віконному інтерфейсі виводяться персональні дані студента – ПІБ, дата народження, номер залікової книжки тощо (див. лаб. роботу 1), але лише при вірно введеному паролі.
3. Виконати компіляцію розробленого 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 (<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 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 побудовано 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.

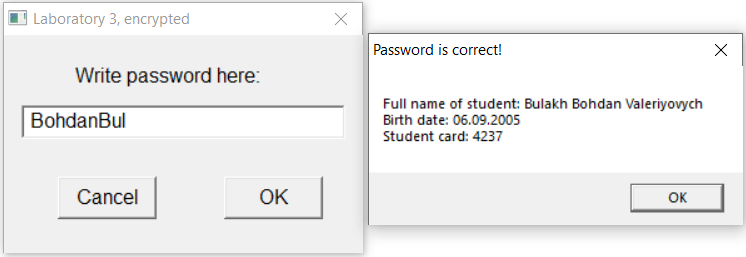
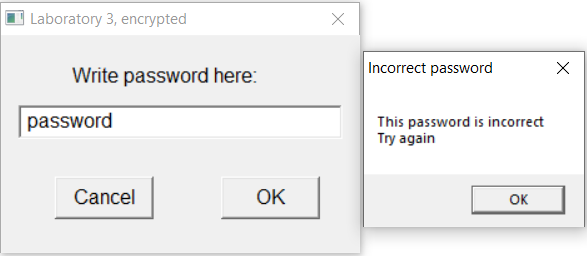
**Виконання роботи**

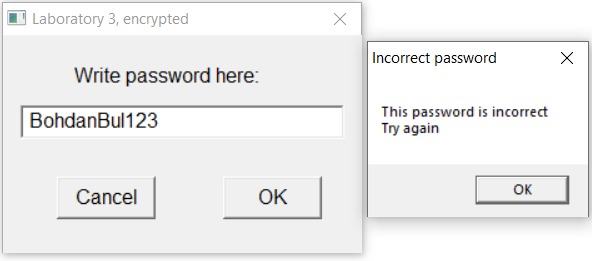
**4.** Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.

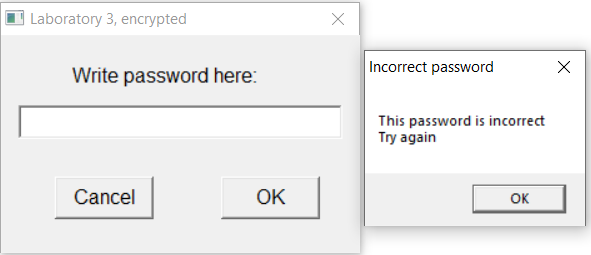
Пароль - **BohdanBul**

* Без шифрування
  + При введенні правильного пароля
  + При введенні неправильного пароля



* Із XOR шифруванням
  + При правильному паролі
  + При неправильному паролі

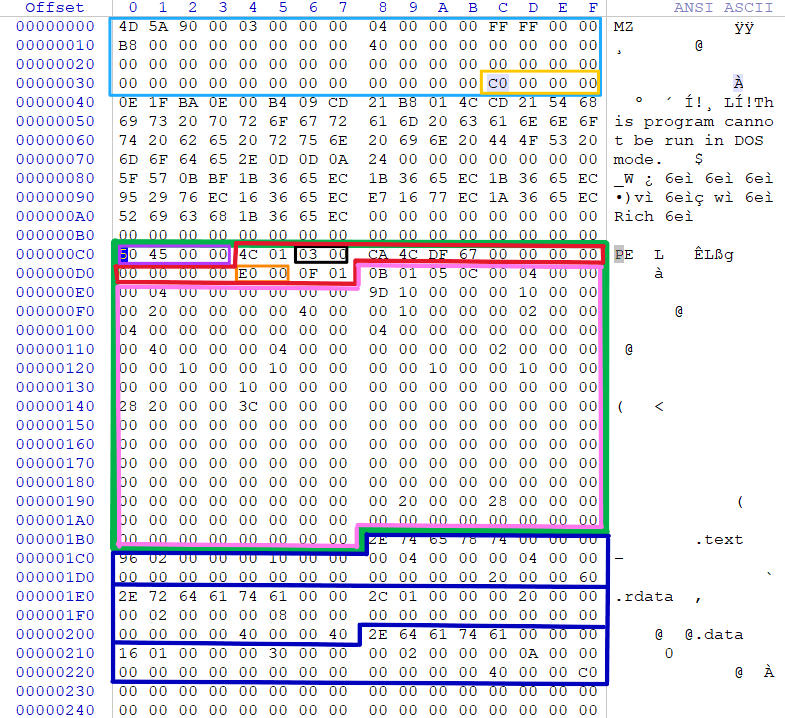




**6.** На скріншоте перших 25 рядків вмісту файлу обвести кольоровим олівцем або фломастером області MS-DOS заголовка (DOS\_HEADER), PE заголовка (PE\_HEADER) і таблиці секцій (SECTION\_HEADERS). Скріншот привести в звіті по лабораторній роботі.

Кольори різних складових частин файлу:

* DOS\_HEADER (64 байти) - блакитний
* Вказівник на початок PE\_HEADER (4 байти) - жовтий, зсув для PE\_HEADER: C0 00 00 00 → 00 00 00 C0 → 192 байти
* PE\_HEADER (248 байтів) - зелений
* PE Signature (4 байти) - бірюзовий
* Заголовок PE-файлу (20 байтів) - червоний
* Кількість Section Headers (2 байти) - чорний, кількість: 03 00 → 00 03 → 3 заголовки
* Розмір Optional Header (2 байти) - помаранчевий, розмір: E0 00 → 00 E0 → 224 байти
* Optional Header (224 байти) - рожевий
* Section Headers (.text, .rdata, .data - по 40 байтів кожен) - синій



**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).

3.3.1. Machine Types

|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Machine type | 01 4C | Intel 386 або новішої версії та сумісні процесори |

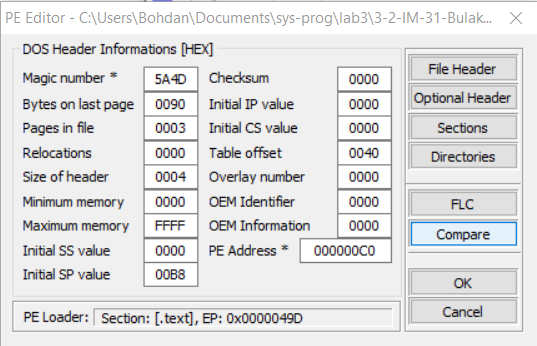
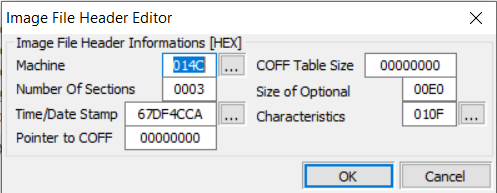
3.4.1. Optional Header Standard Fields (Image Only)

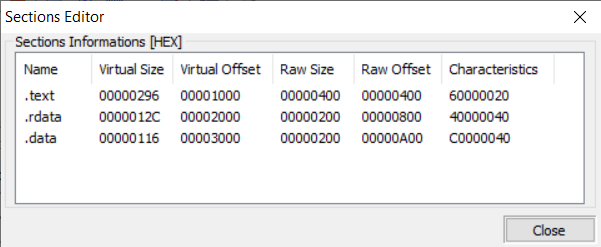
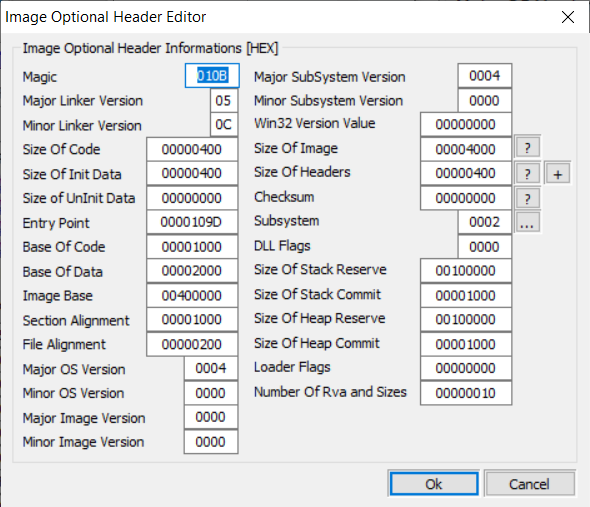
|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Magic | 01 0B | Беззнакове ціле число, що ідентифікує стан файлу. 0x10B - ідентифікує його як звичайний виконуваний файл |
| MajorLinkerVersion | 05 | Номер основної версії компонувальника |
| MinorLinkerVersion | 0C | Номер другорядної версії компонувальника |
| SizeOfCode | 00 00 04 00 | Розмір секції коду або сума всіх секцій коду, якщо їх є декілька |
| SizeOfInitializedData | 00 00 04 00 | Розмір ініціалізованої секції даних або сума всіх таких секцій, якщо їх декілька |
| SizeOfUninitializedData | 00 00 00 00 | Розмір секції неініціалізованих даних (BSS), або сума всіх таких секцій, якщо є декілька секцій BSS |
| AddressOfEntryPoint | 00 00 10 9D | Адреса вхідної точки |
| BaseOfCode | 00 00 10 00 | Адреса початку секції коду |
| BaseOfData | 00 00 20 00 | Адреса початку секції змінних |

4. Section Table (Section Headers)  
 .data section

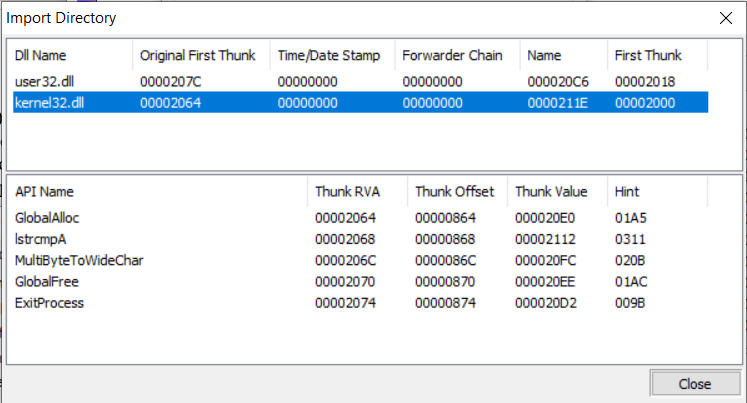
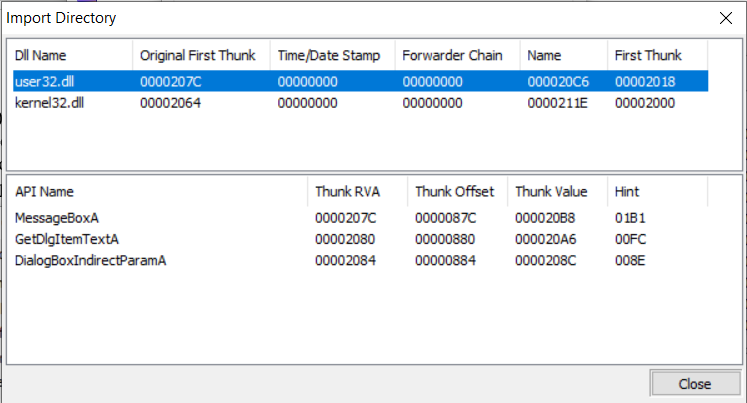
|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Name | .data | 8-байтовий рядок ASCII із доповненням нульовими значеннями |
| VirtualSize | 00 00 01 16 | Повний розмір секції при завантаженні в пам'ять |
| VirtualAddress | 00 00 30 00 | Адреса першого байту секції |
| SizeOfRawData | 00 00 02 00 | Розмір секції |
| PointerToRawData | 00 00 0A 00 | Вказівник на першу сторінку розділу в COFF файл, зсув для секції з даними |
| PointerToRelocations | 00 00 00 00 | Вказівник файлу на початок записів переміщення для розділу (0 для виконуваних файлів) |
| PointerToLinenumbers | 00 00 00 00 | Вказівник файлу на початок записів номерів рядків для розділу |
| NumberOfRelocations | 00 00 | Кількість релокаційних входжень в розділ (0 для виконуваних файлів) |
| NumberOfLinenumbers | 00 00 | Кількість номерних записів розділу |
| Characteristics | C0 00 00 40 | Прапори, що описують характеристики розділу |

**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.



Інформація, знайдена у файлі вручну, відповідає отриманій за допомогою **PE Tools.**

**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 побудовано 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.

Знаходимо секцію з даними:

Зеленим позначено поле, з якого визначаємо зсув:

00 0A 00 00 → 00 00 0A 00

Знаходимо в програмі:

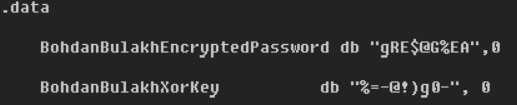
**12.** Виконати шифрування пароля за допомогою функції XOR, знову скомпілювати EXE -файл і переконатися, що тепер вони не виявляються явним чином в тексті виконуваного EXE - файлу. Привести скріншоти цієї програми в режимах «Hex» і «Text» в звіті по лабораторній роботі.

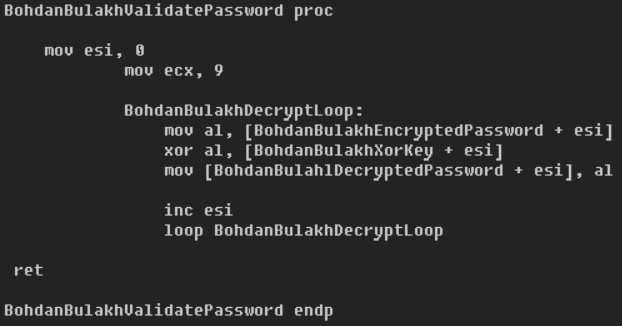
Пароль: **BohdanBul**

Ключ для шифрування XOR: **%=-@!)g0-**

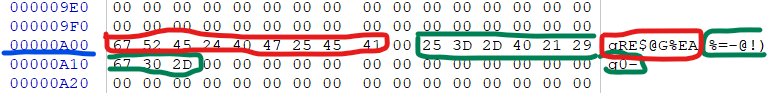
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Текст | Двійкове | | | | | | | | |
| BohdanBul | 01000010 | 01101111 | 01101000 | 01100100 | 01100001 | 01101110 | 01000010 | 01110101 | 01101100 |
| %=-@!)g0- | 00100101 | 00111101 | 00101101 | 01000000 | 00100001 | 00101001 | 01100111 | 00110000 | 00101101 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Результат **XOR** | | | | | | | | | |
| gRE$@G%EA | 01100111 | 01010010 | 01000101 | 00100100 | 01000000 | 01000111 | 00100101 | 01000101 | 01000001 |

Зашифрований пароль та код для XOR у тексті програми:

Використання XOR у програмі:

Зашифрований пароль та ключ для XOR в exe файлі:

**Висновки**

Як можна побачити з лабораторної роботи, зберігати паролі у вигляді звичайного тексту всередині програми ненадійно, оскільки їх можна з легкістю знайти під час перегляду виконуваного файлу у hex-редакторі.

Xor шифрування дозволяє зберігати пароль у зашифрованому вигляді. Його все ще можна відновити, оскільки ключ також зберігається в програмі, але цей спосіб вже забезпечує мінімальний захист.