**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»

Виконав: Перевірив:

Студентка 2-го курсу групи ІМ-21 Павлов В.Г.

Рабійчук Дар’я Олександрівна   
номер у списку групи: 18

Київ 2024

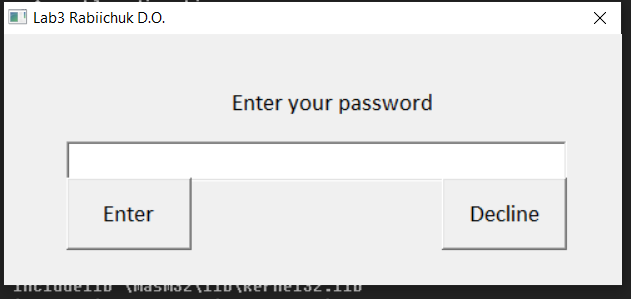
**Мета роботи:** ознайомлення із специфікацією COFF (Common Object File Format). Вивчення прийомів дослідження структури файлів PE-формату.

**Виконання роботи:**

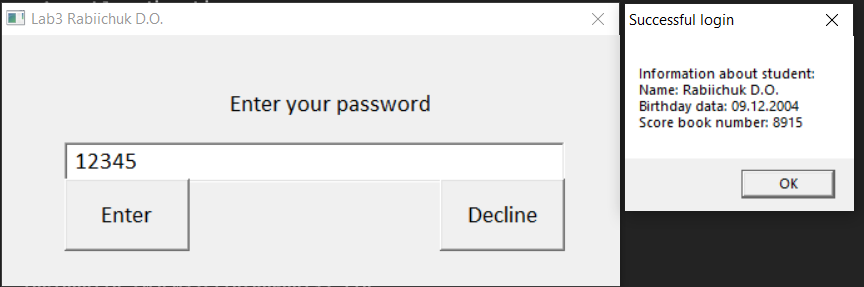
**4. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.**

**Тестування першої програми без шифрування:**

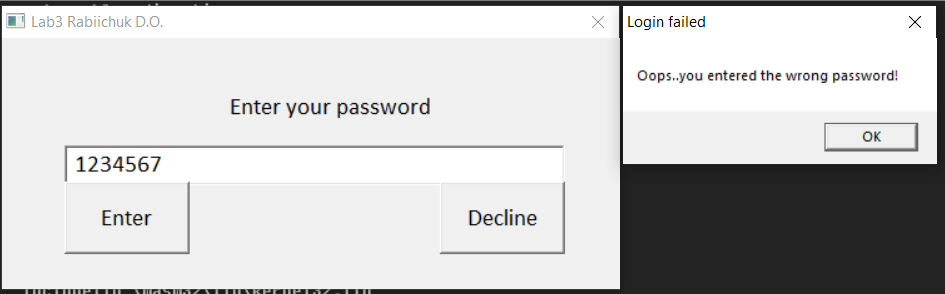
Вікно для введення паролю:



Введення правильного паролю:

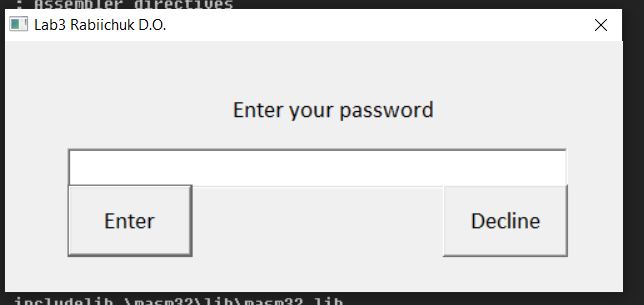


Введення неправильного паролю:

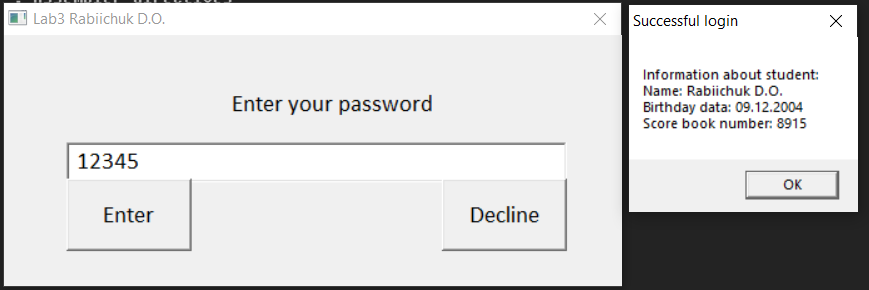


**Тестування першої програми з шифруванням:**

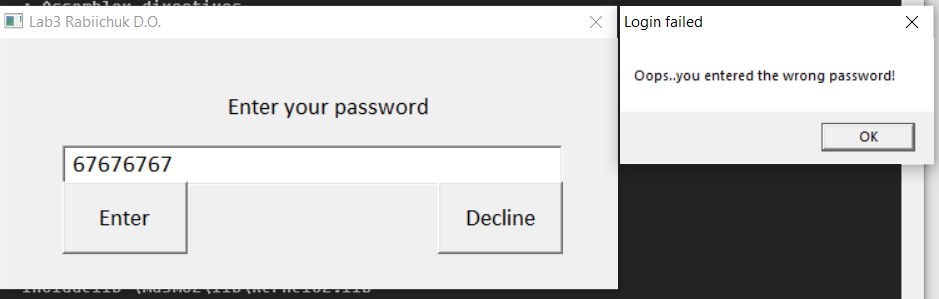
Вікно для введення паролю:



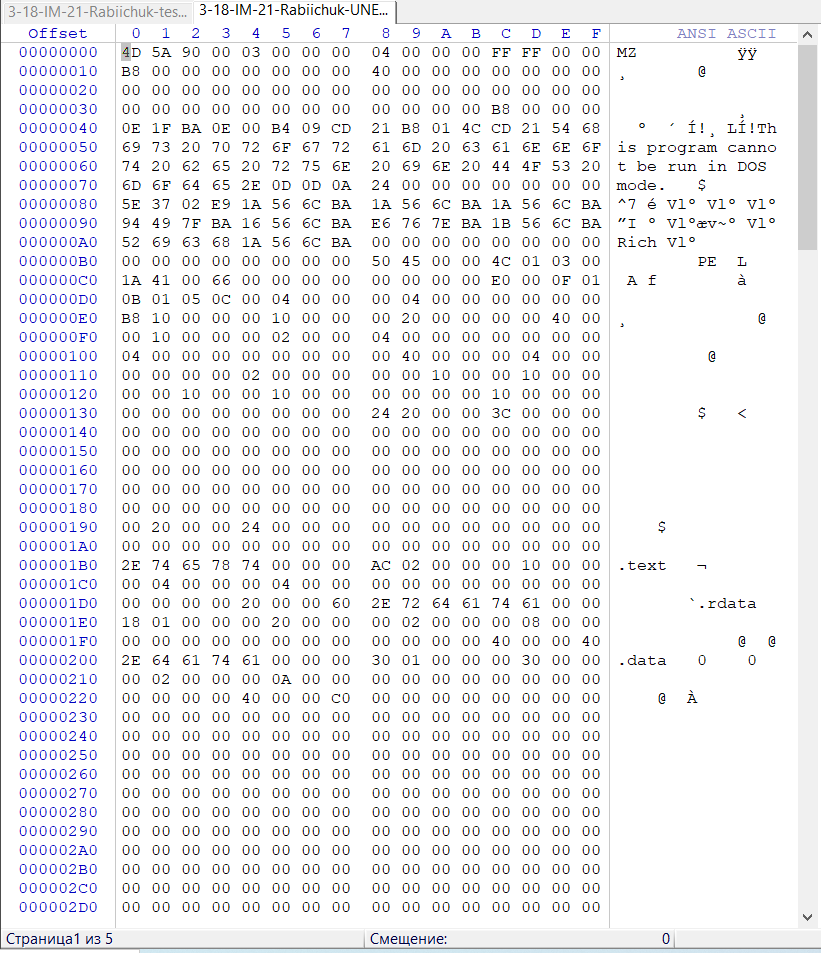
Введення правильного паролю:



Введення неправильного паролю:



**5-6. Дослідження файлу в WinHex:**

****

**Пояснення до виділених фрагментів:**

Зеленим кольором виділено 64 байти DOS Header, в останніх його 4 байтах блакитним кольором виділено вказівник на PE Header. PE Signature обведений жовтогарячим кольором має розмір у 4 байти, в якому записано байтову сигнатуру “PE\0\0”. Після PE Signature у файлі міститься заголовок PE-файлу, який також має сталу довжину, а саме 20 байтів. На скріншоті він виділений червоним кольором. Після нього йде Optional Header виділений темно-червоним кольором. Його розмір міститься в заголовку PE-файлу де він займає 2 байти, на фото обведений жовтим кольором. Section Headers йде одразу після Optional Header. Він обведений **чорним** кольором. Кількість секцій у цьому заголовку записано в Optional Header, де ці дані займають 2 байти (обведено фіолетовим кольором).

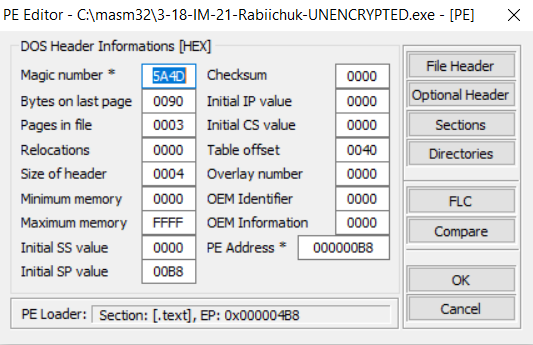
**7-8. Таблиці з параметрами файлів:**

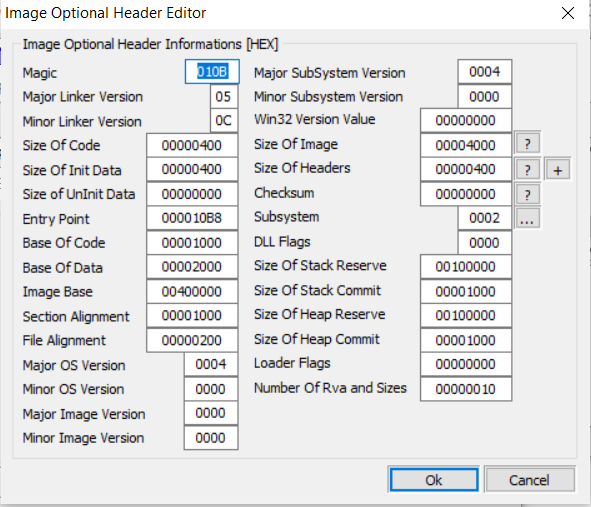
|  |  |  |
| --- | --- | --- |
| **Розділ 3.3.1: Machine Types** | | |
| Поле | Значення | Опис |
| Machine type | 01 4С | Тип машини: *Intel 386* або пізніше або інші схожі процесори |

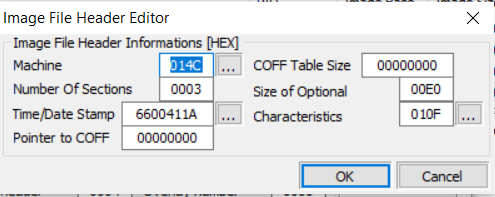
|  |  |  |
| --- | --- | --- |
| **Розділ 3.4.1: Optional Header Standard Fields (Image Only)** | | |
| Поле | Значення | Опис |
| Magic | 01 0В | Ідентифікатор нормального файлу типу .ехе |
| Major Linker Version | 05 | Номер головної версії компонувальника |
| Minor Linker Version | 0С | Номер другорядної версії компонувальника. |
| Size Of Code | 00 00 04 00 | Розмір секції коду |
| Size Of Initialized Data | 00 00 04 00 | Розмір ініціалізованого розділу даних |
| Size Of Uninitialized Data | 00 00 00 00 | Сума розмірів всіх секцій, що містять неініціалізовані дані |
| Address Of Entry Point | 00 00 10 00 | Адреса вхідної точки |
| Base Of Code | 00 00 10 00 | Адреса початку розділу коду |
| Base Of Data | 00 00 20 00 | Адреса початку розділу змінних |

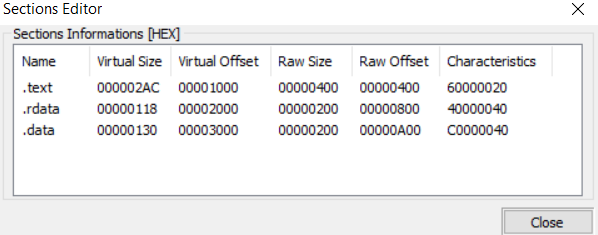
|  |  |  |
| --- | --- | --- |
| **Розділ 4: Section Table (Section Headers)** | | |
| Поле | Значення | Опис |
| Name | .data\0\0\0 | Назва секції - 8 бітний стрінг ASCII |
| Virtual Size | 00 00 01 42 | Загальний розмір розділу при завантаженні в пам'ять |
| Virtual Address | 00 00 40 00 | Адреса першого байту секції |
| Size Of Raw Data | 00 00 02 00 | Розмір секції |
| Pointer To Raw Data | 00 00 20 00 | Вказівник на першу сторінку секції в COFF файл |
| Pointer To Relocations | 00 00 00 00 | Вказівник файлу на початок записів переміщення для секції |
| Pointer To Linenumbers | 00 00 00 00 | Вказівник файлу на початок записів номерів рядків для секції |
| Number Of Relocations | 00 00 | Кількість релокаційних входжень в розділ |
| Number Of Linenumbers | 00 00 | Кількість номерних записів секціх |
| Characteristics | 40 00 00 40 | Прапори опису характеристики секції |

**9. Аналіз того ж файлу за допомогою меню PE Tools:**

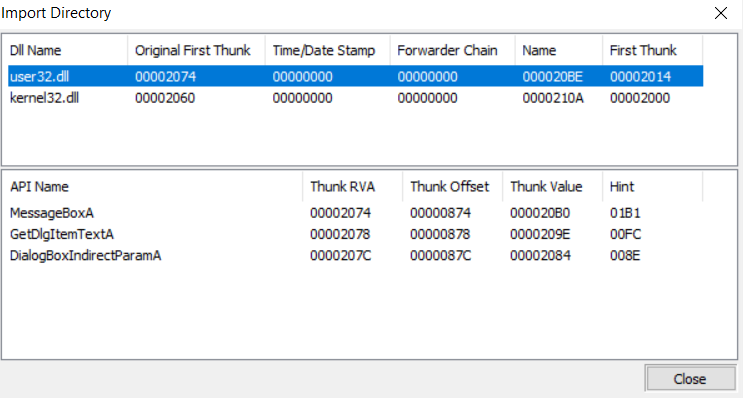
****

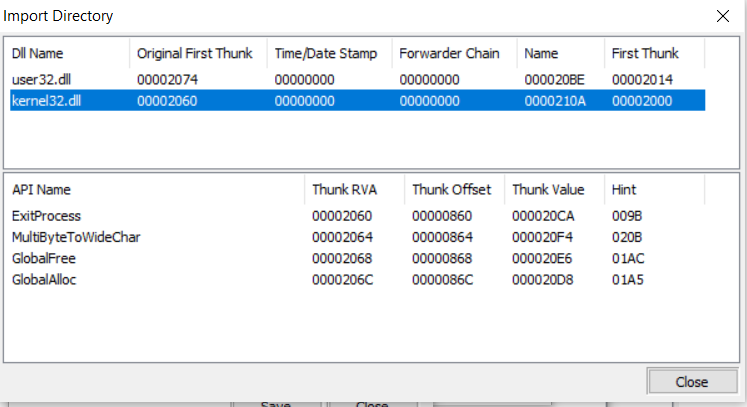
****

****

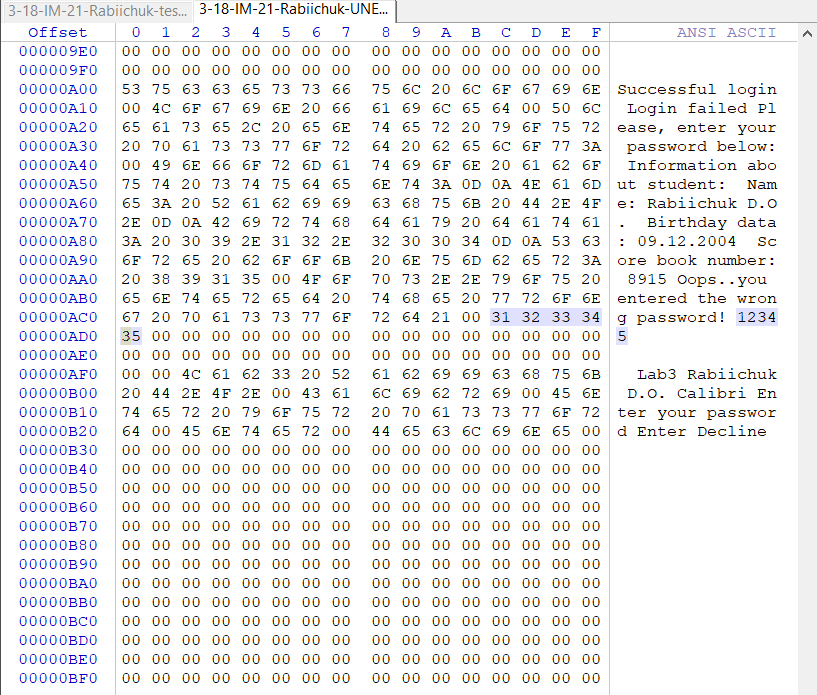
****

**10.** **Дослідити таблицю імпорту (Import Directory) даного файлу і визначити, які саме функції використовуються з бібліотек, що підключаються.**

****

****

**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.**

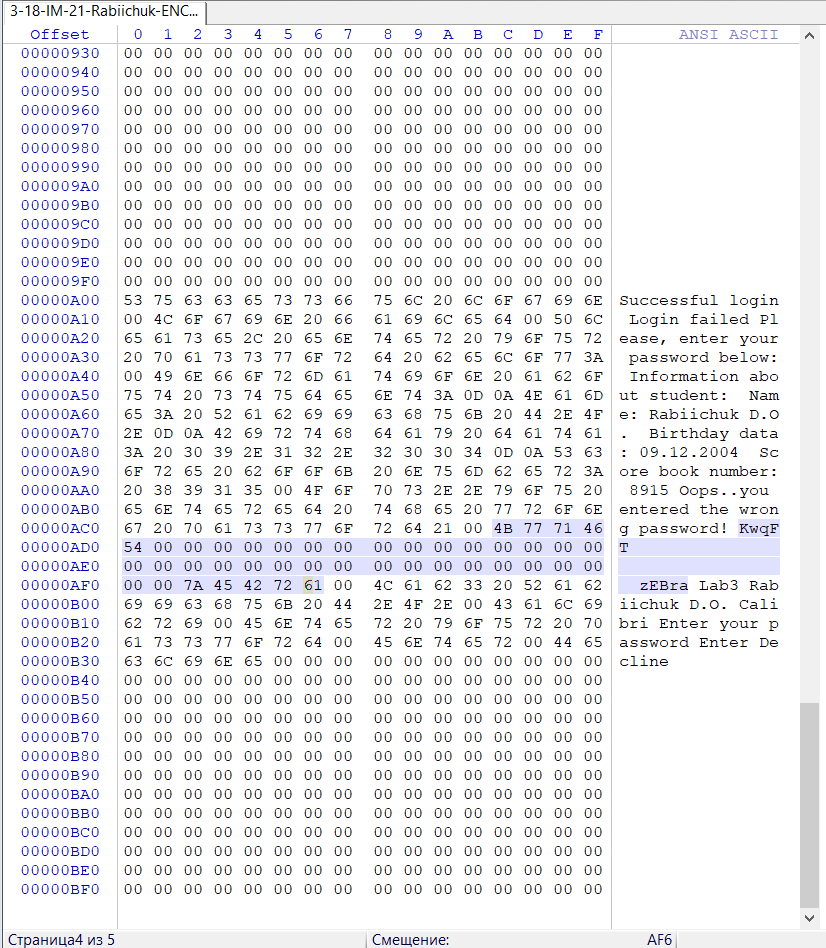
****

**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.**

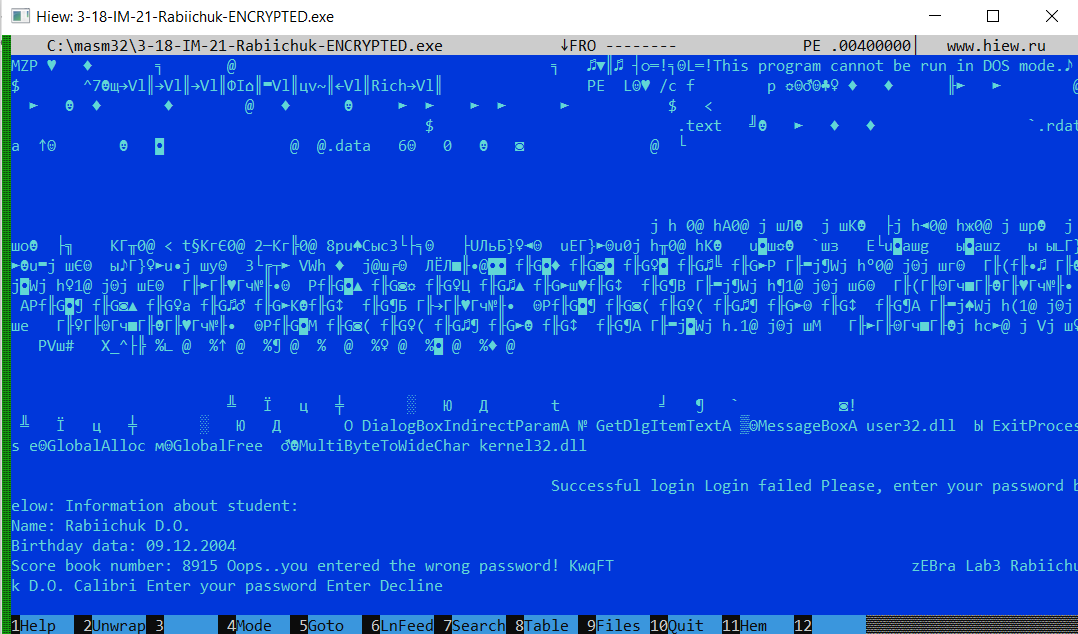
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ASCII** | **Binary** | | | | |
| 12345 | 00110001 | 00110010 | 00110011 | 00110100 | 00110101 | |
| zEBra | 01111010 | 01000101 | 01000010 | 01110010 | 01100001 | |
| **XOR** | | | | | |
| KwqFT | 01001011 | 01110111 | 01110001 | 01000110 | 01010100 | |

**Вміст файлу програми з шифруванням:**

Режим «Hex»

****

Режим «Text»



**Висновки:**

При виконанні цієї лабораторної роботи я ознайомилася з структурою файлів з розширенням .exe , а саме з розташуванням та наповненням двох частин файлів такого формату – заголовка та модуля завантаження. Також навчилася глибоко досліджувати параметри файлу за допомогою програми PE Tools, WinHex та Hiew32. X

Також я навчилась методу шифрування даних XOR. Під час даної роботи було написано 2 програми, у яких при введені правильного паролю ми могли побачити деякі дані про студента, а при введені неправильного – отримували повідомлення про помилку. В першій програмі наш пароль зберігався у самому коді, а у другій ми робили шифрування. Звичайно, що метод XOR є не самим надійним, але на відміну від першого варіанту зберігання паролю його не так легко дізнатись.