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 II курсу ФІОТ доц. Павлов В. Г. групи ІМ-22 Басараб Станіслав Анатолійович  
номер у списку групи: 2

Київ 2024

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

**Порядок виконання роботи**

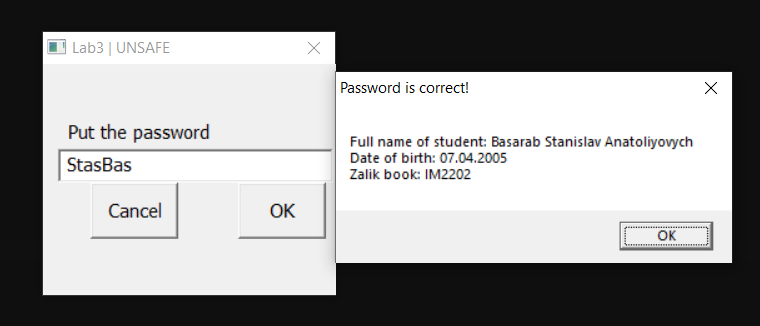
1. Вивчити структуру програм формату **EXE**.
2. Розробити програму на мові Асемблер, за допомогою якої у віконному інтерфейсі виводяться персональні дані студента – ПІБ, дата народження, номер залікової книжки тощо (див. лаб. роботу 1), але лише при вірно введеному паролі.
3. Виконати компіляцію розробленого 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 версія\*).
6. На скріншоті 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 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\*). Всі 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 оригіналу 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.

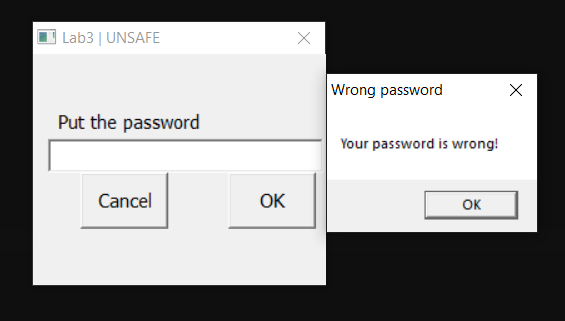
**Виконання роботи**

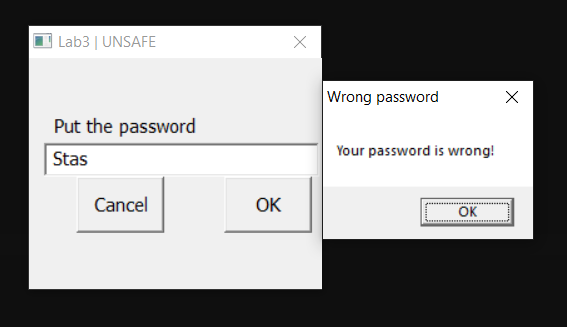
Перевірка роботи програми шляхом введення як правильного, так і невірного паролів.

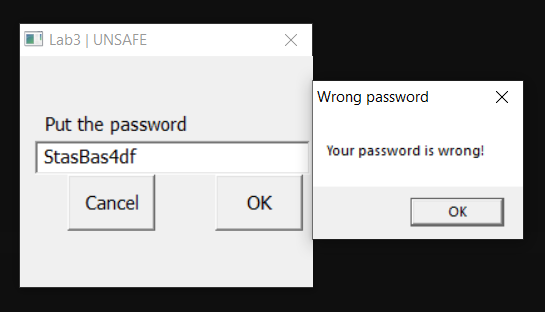
Правильний пароль – StasBas.

* без шифрування

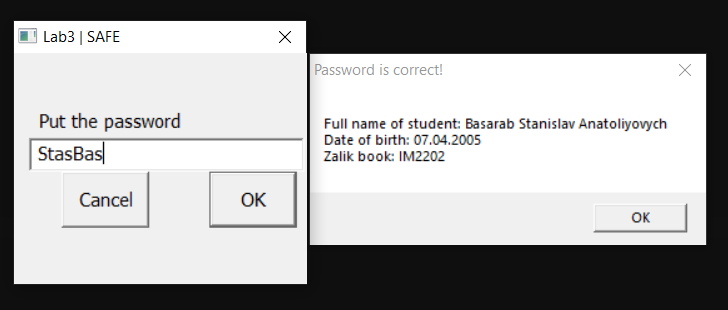


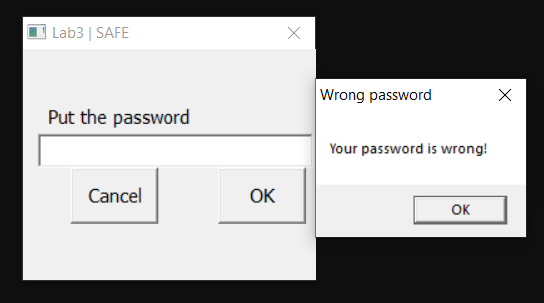


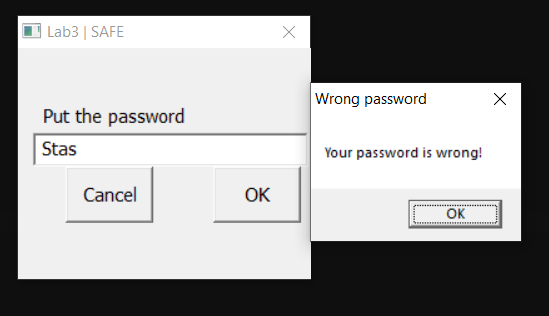


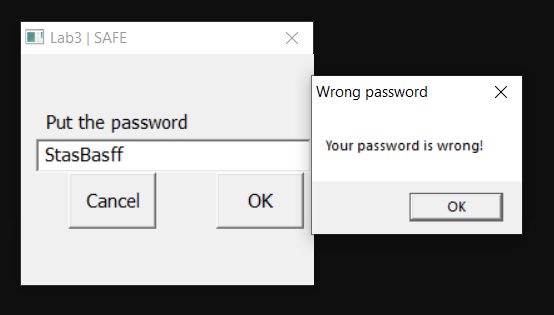


* з шифруванням







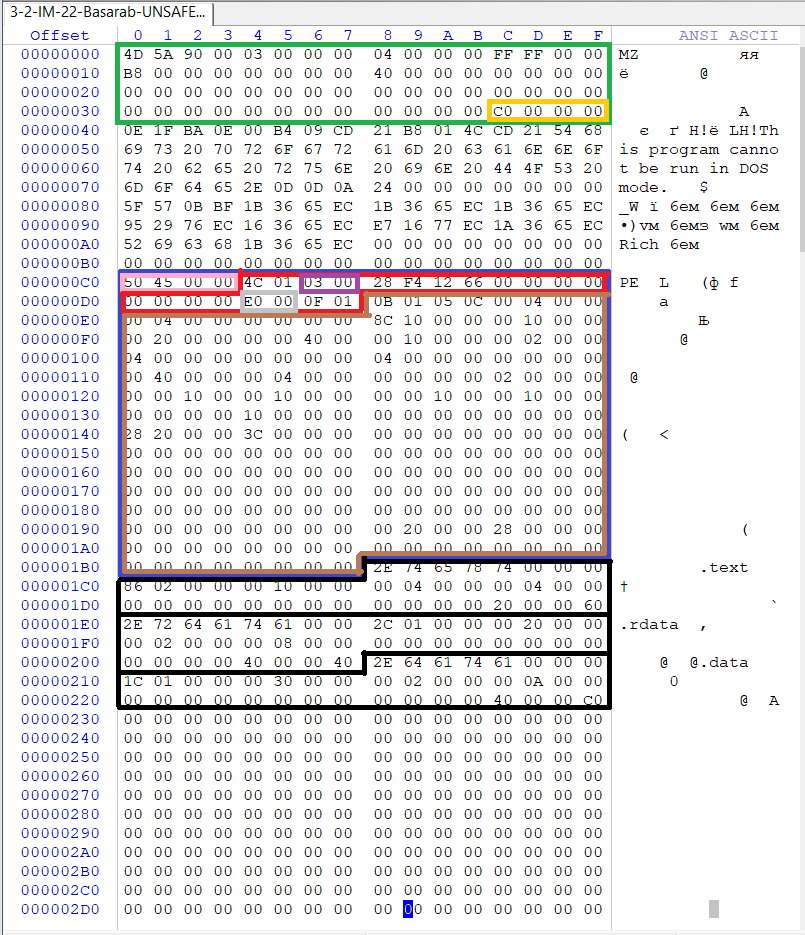


Дослідження виконавчого файлу за допомогою **WinHex**

На скріншоті представлені DOS\_HEADER (зелений колір), PE\_HEADER (синім кольором) та SECTION\_HEADERS (чорним кольором). Об’єм DOS\_HEADER – 64 байти, з яких останні 4 (жовтий колір) містять вказівник на початок PE\_HEADER. Тож зсув для PE\_HEADER = 00 00 00 С0 (192 байти).

На початку PE\_HEADER знаходиться PE Signature (рожевий колір) розміром 4 байти з сигнатурою “PE\0\0”. Далі червоним кольором позначено заголовок PE-файлу розміром 20 байтів. Одразу після заголовоку PE-файлу представлений Optional Header розміром, визначеним у заголовку PE-файлу (сірим кольором, 2 байти). Тож значення 00 E0h (224 байти).

Section Headers на скріншоті позначені чорним кольором, а інформація про кількість секцій обведена фіолетовим кольором (тобто всього три секції). А саме: .text, .rdata, .data, розміром по 40 байтів кожна.



Таблиці параметрів файла

3.3.1. Machine Types

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

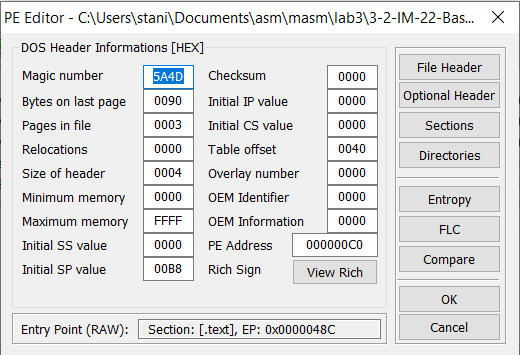
3.4.1. Optional Header Standard Fields (Image Only)

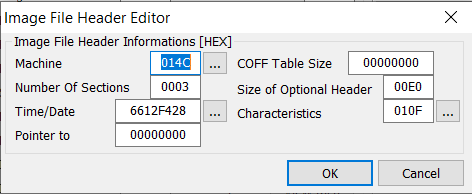
|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Magic | 01 0В | Беззнакове ціле число, що ідентифікує стан  файлу. 0x10B - ідентифікує його як звичайний виконуваний файл |
| MajorLinkerVersion | 05 | Номер основної версії компонувальника |
| MinorLinkerVersion | 0С | Номер другорядної версії 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 8C | Адреса вхідної точки |
| BaseOfCode | 00 00 10 00 | Адреса початку секції коду |
| BaseOfData | 00 00 20 00 | Адреса початку секції змінних |

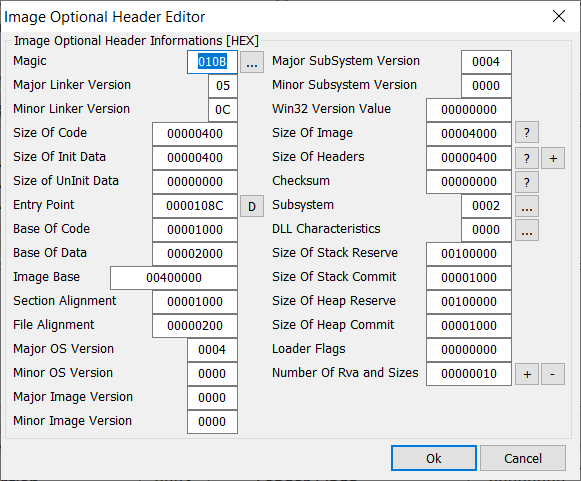
4. Section Table (Section Headers)  
.data section

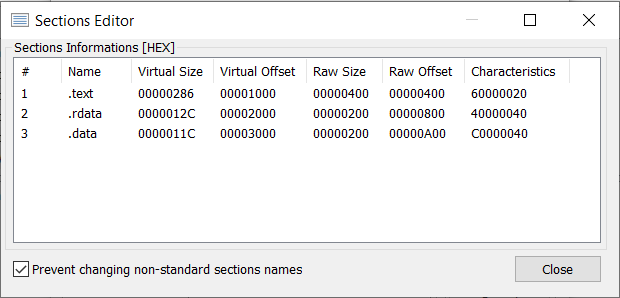
|  |  |  |
| --- | --- | --- |
| Поле | Значення | Опис |
| Name | .data\0\0\0 | 8-байтовий рядок ASCII із доповненням нульовими значеннями |
| VirtualSize | 00 00 01 1C | Повний розмір секції при завантаженні в пам'ять |
| VirtualAddress | 00 00 30 00 | Адреса першого байту секції |
| SizeOfRawData | 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 |

Дослідження файлу за допомогою меню **"PE Editor"** програми **PE Tools**



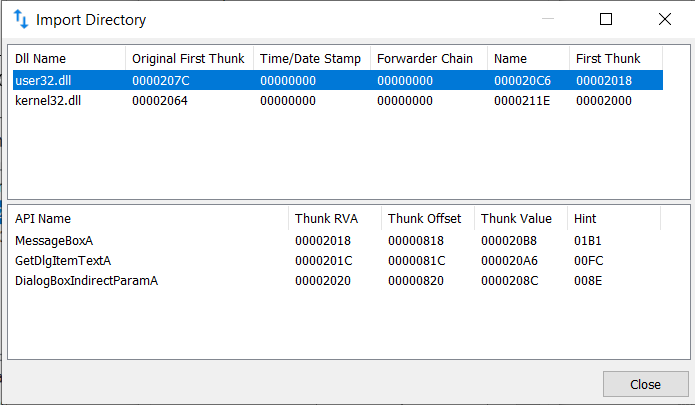


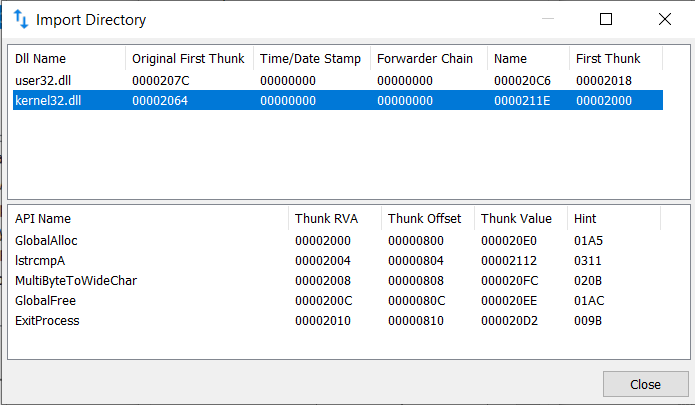




З проведеного вище дослідження можна стверджувати, що дані, занесені до таблиць, відповідають даним, отриманим під час дослідження в програмі **PE Tools.**

Дослідження таблиці імпорту (**Import Directory**) файлу і визначення, які саме функції використовуються з бібліотек, що підключаються.

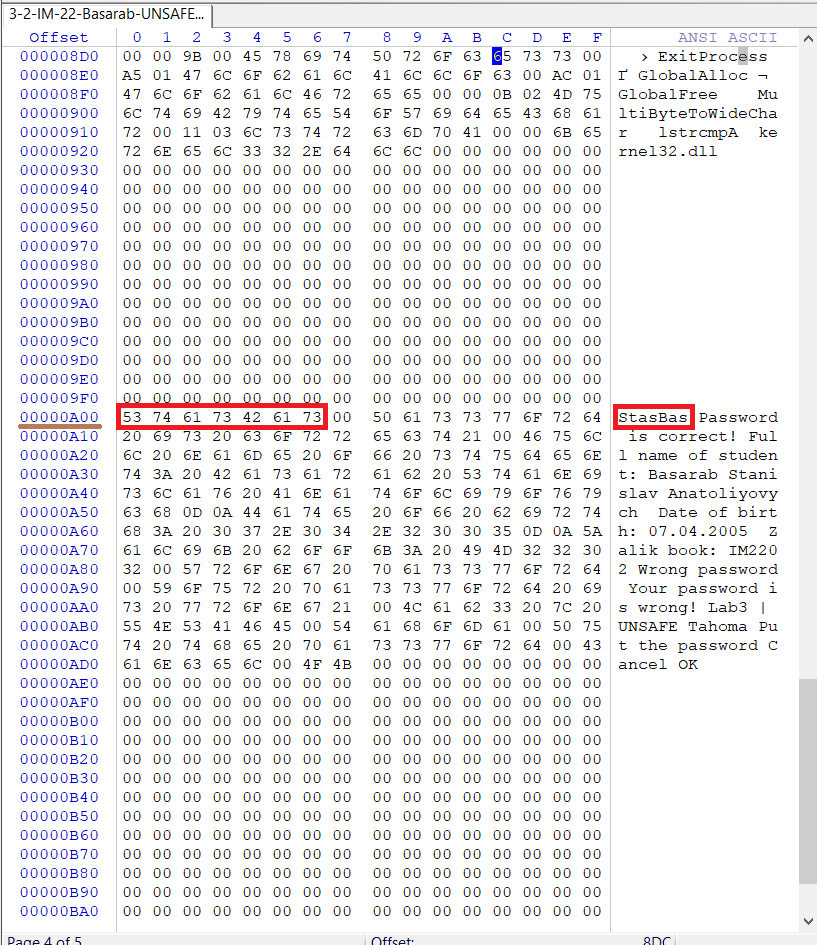




З скріншотів видно, що в програму імпортуються файли бібліотек user32.dll, kernel32.dll, по 3 та 5 функцій відповідно.

Пошук тексту оригінала паролю, що міститься в тексті програми за допомогою **HEX**-редактора.

За зсувом (підкреслено) з таблиці знаходимо секцію з даними. Знаходимо текст паролю.



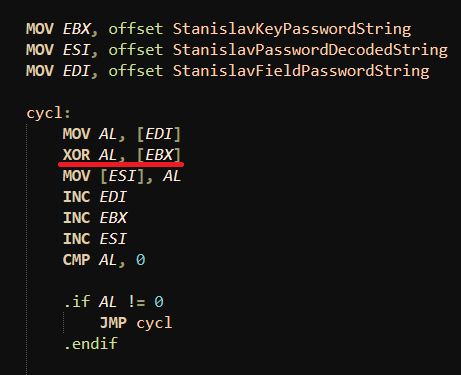
Шифрування пароля за допомогою функції XOR

Зашифруємо пароль за допомогою ключа: «5--4552»

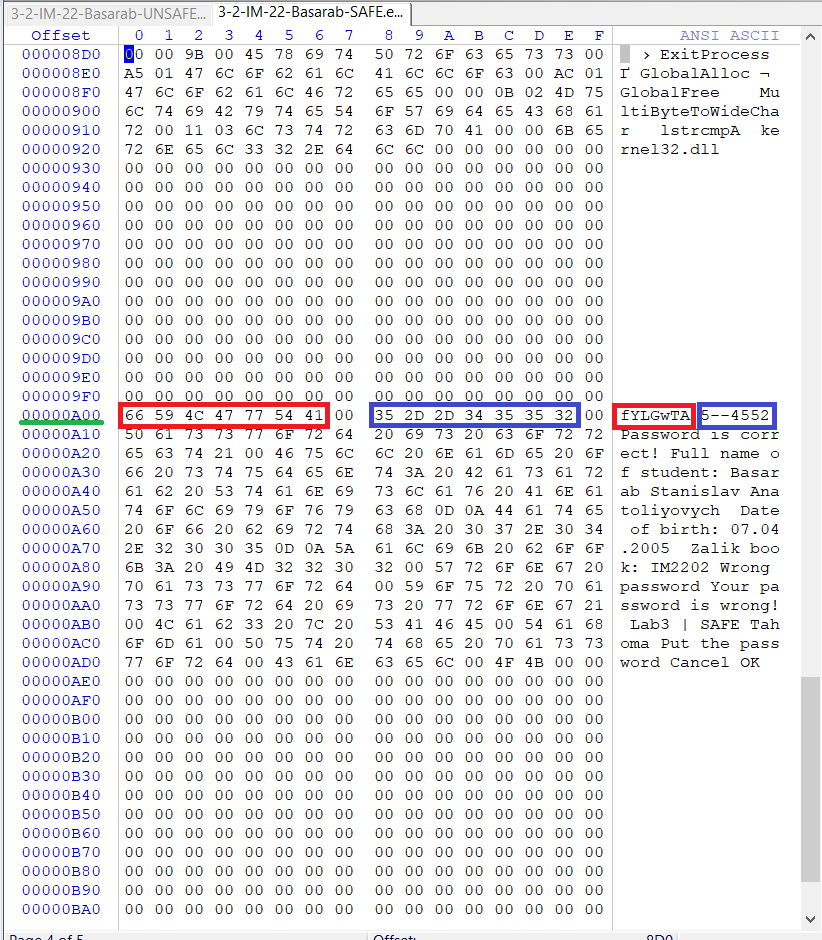
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ASCII | Binary | | | | | | |
| StasBas | 01010011 | 01110100 | 01100001 | 01110011 | 01000010 | 01100001 | 01110011 |
| 5--4552 | 00110101 | 00101101 | 00101101 | 00110100 | 00110101 | 00110101 | 00110010 |
| Результат після операції XOR | | | | | | | |
| fYLGwTA | 01100110 | 01011001 | 01001100 | 01000111 | 01110111 | 01010100 | 01000001 |

Зашифрований пароль: fYLGwTA

Операція **xor** у коді програми:



Знову ж таки знаходимо по зсуву секцію з даними уже в новій програмі. Можна побачити, що оригінального паролю немає, натомість є зашифрований пароль та ключ, щоб розшифровувати пароль, введений користувачем.



**Висновок**

Отже, під час виконання лабораторної роботи проведено ознайомлення з специфікацією COFF (Common Object File Format), досліджено структуру файлів PE-формату за допомогою програм **WinHex** та **PE Editor.** Відповідно до специфікації складено таблицю параметрів файла. Значення цих параметрів порівняно зі значеннями з програми **PE Editor.** Ці дані повністю збігаються. Досліджено таблицю імпорту, а також які саме функції використовуються з бібліотек, що підключаються.  
 Порівняно між собою дві програми (одну з шифруванням паролю, іншу - без). Звичайно, що метод xor-шифрування не є надійним, проте за допомогою нього ми не можемо явно відслідкувати оригінальний пароль прямо у виконуваному файлі програми, на відміну від програми без шифрування. І все ж таки у програмі з шифруванням ми можемо бачити ключ та зашифрований пароль, що створює додатковий ризик «взлому» програми. Тому слід обирати надійніші способи шифрування паролю.