**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»

Варіант 21

Виконав: Перевірив:

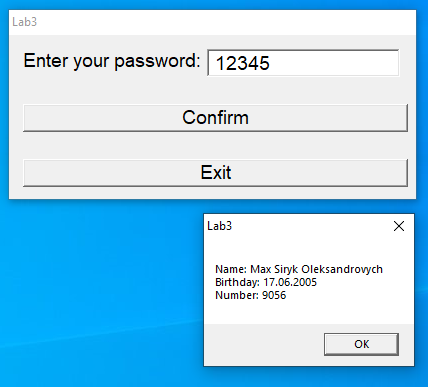
студент групи ІМ-21 доцент Павлов В.Г.

Сірик Максим Олександрович

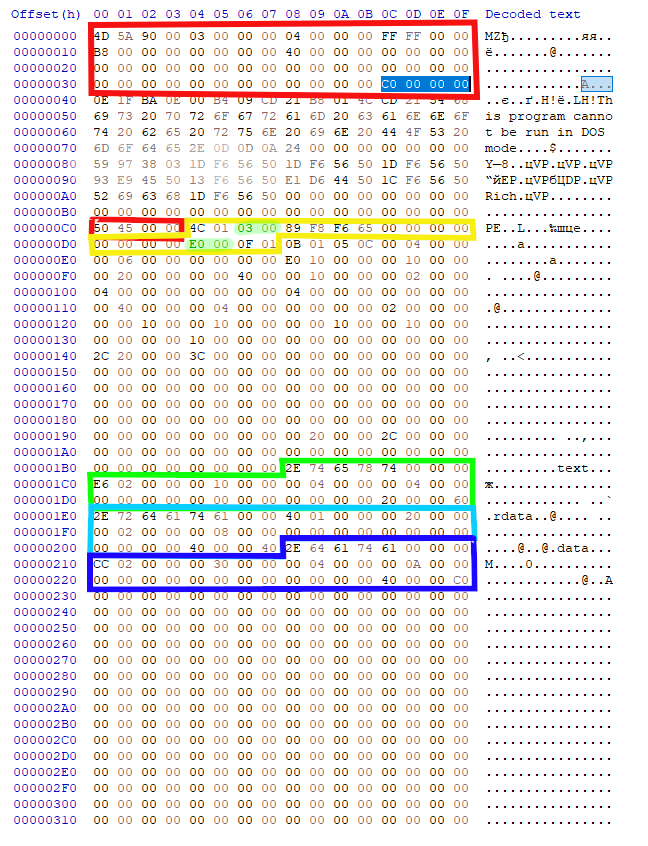
Номер у списку групи: 21

**Київ 2024**

# Скриншоти виконання програми



# Дослідити виконавчий файл

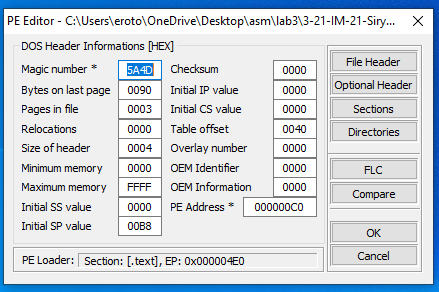


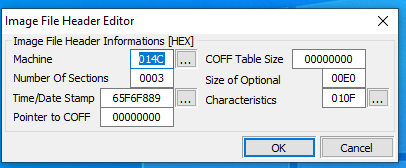
|  |  |  |
| --- | --- | --- |
| Image Dos Header ○Red1 | | |
| Field | Value | Description |
| Magic number (2) | 0x5A4D | Offset 0x00; MZ |
| PE address pointer (4) | 0x00 00 00 C0 | Offset 0x3C; PE start |
|  | | |
| NT headers | | |
| Image NT signature (4) ○Red2 | 0x00 00 45 50 | Offset = PE start = 0x00 00 00 C0  PE00 |
| Image File Header (PE header) ○Yellow | | |
| Machine (2) | 0x01 4C | Offset = PE start + 0x04;  Intel 386 or later cpu type (Тип процесора cpu) |
| Number of sections (2) | 0x00 03 | Offset = PE start + 0x06;  we have 3 sections |
| Size of optional headers (2) | 0x00 E0 | Offset = PE start + 0x14;  224 bytes of optional headers |
| Optional Header | | |
| Magic (2) | 0x01 0B | Offset = PE start + 24;  Формат виконавчого файлу  PE32 format (not PE32+) |
| Major Linker Version (1) | 0x05 | Offset = PE start + 24 + 2;  Linker major version number. (Головна версія лінкера) |
| Minor Linker Version (1) | 0x0C | Offset = PE start + 24 + 3;  Linker minor version number. (Другорядна версія лінкера) |
| Size of code (4) | 0x00 00 04 00 | Offset = PE start + 24 + 4;  The size of the code (text) section, or the sum of all code sections if there are multiple sections. (Розмір секції коду) |
| Size of initialized data (4) | 0x00 00 06 00 | Offset = PE start + 24 + 8;  The size of the initialized data section, or the sum of all such sections if there are multiple data sections. (Розмір секції з ініціалізованими даними) |
| Size Of Uninitialized Data (4) | 0x00 00 00 00 | Offset = PE start + 24 + 12;  Size of the uninitialized data section (BSS), or the sum of all such sections if there are multiple BSS sections. (Розмір секції з неініціалізованими даними |
| Entry Point (4) | 0x00 00 10 E0 | Offset = PE start + 24 + 16;  The address of the entry point relative to the image base when the executable file is loaded into memory. For program images, this is the starting address. For device drivers, this is the address of the initialization function. An entry point is optional for DLLs. When no entry point is present, this field must be zero. (Адрес точки входу віднисно image base) |
| Base Of Code (4) | 0x00 00 10 00 | Address, relative to image base, of beginning of code section, when loaded into memory. (Адреса початку секції коду, коли завантажена в память відносно image base) |
| Base Of Data (4) | 0x00 00 20 00 | Address, relative to image base, of beginning of data section, when loaded into memory.  (Адреса початку секції даних, коли завантажена в память відносно image base) |
| Image Base (4) | 0x00 40 00 00 | Offset = PE start + 24 + 28;  The preferred address of the first byte of image when loaded into memory; must be a multiple of 64 K. The default for DLLs is 0x10000000. The default for Windows CE EXEs is 0x00010000. The default for Windows NT, Windows 2000, Windows XP, Windows 95, Windows 98, and Windows Me is 0x00400000. (Алреса першого байту image, коли завантажений у память) |
| Section alignment (4) | 0x00 00 10 00 | Offset = PE start + 24 + 32;  The alignment (in bytes) of sections when they are loaded into memory. It must be greater than or equal to FileAlignment. The default is the page size for the architecture. (вирівнювання секцій, коли вони завантажені у память) |
| File Alignment (4) | 0x00 00 02 00 | Offset = PE start + 24 + 36;  The alignment factor (in bytes) that is used to align the raw data of sections in the image file. The value should be a power of 2 between 512 and 64 K, inclusive. The default is 512. If the SectionAlignment is less than the architecture's page size, then FileAlignment must match SectionAlignment. (вирівнбвання що використовується для raw data в image file) |
|  | | |
| Image Section Header |  |  |
| Name (8) | .text | Offset = PE start + 24 + Size of optional headers(224) = StartText (назва секції) |
| Virtual Size (4) | 0x00 00 02 E6 | Offset = StartText + 8  The total size of the section when loaded into memory. (загальний розмір секції, коли завантажено у память) |
| Virtual address (4) | 0x00 00 10 00 | Offset = StartText + 12  The address of the first byte of the section relative to the image base when the section is loaded into memory. (адреса першого байту секції відносно image base) |
| Raw size (SizeOfRawData) (4) | 0x00 00 04 00 | Offset = StartText + 16; (Розмір секції) |
| Raw offset (PointerToRawData) (4) | 0x00 00 04 00 | Offset = StartText + 20;  Offset where we would read data (Вказівник на секцію у файлі Coff) |
| Pointer To Relocations (4) | 0x00 00 00 00 | Offset = StartText + 24;  File pointer to beginning of relocation entries for the section. Set to 0 for executable images or if there are no relocations. (Вказівник на початок переесення входів у секції) |
| Pointer To Line numbers (4) | 0x00 00 00 00 | Offset = StartText + 28;  File pointer to beginning of line-number entries for the section. Set to 0 if there are no COFF line numbers. (Вказівник на початок секції з рядками чисел) |
| Number Of Relocations (2) | 0x00 00 | Offset = StartText + 32;  Number of relocation entries for the section. Set to 0 for executable images. (кількість перенесення входів у секції) |
| Number Of Line numbers (2) | 0x00 00 | Offset = StartText + 34;  Number of line-number entries for the section. (Кількість записів номерів рядків розділу.) |
| Characteristics (4) | 0x60 00 00 20 | Offset = StartText + 36;  Flags describing section’s characteristics. (Прапорці що описують характеристики секцій) |
|  | | |
| Name | .rdata | Offset = StartText + 40 = StartRData |
| Virtual Size (4) | 0x00 00 01 40 | Read about this section in the .text |
|  | 0x00 00 20 00 | Read about this section in the .text |
| Raw size (SizeOfRawData) (4) | 0x00 00 02 00 | Read about this section in the .text |
| Raw offset (PointerToRawData) (4) | 0x00 00 08 00 | Read about this section in the .text |
| Pointer To Relocations (4) | 0 | Read about this section in the .text |
| Pointer To Line numbers (4) | 0 | Read about this section in the .text |
| Number Of Relocations (2) | 0 | Read about this section in the .text |
| Number Of Line numbers (2) | 0 | Read about this section in the .text |
| Characteristics (4) | 0x40 00 00 40 | Read about this section in the .text |
|  | | |
| Name | .data | Offset = StartRData + 40 |
| Virtual Size (4) | 0x00 00 02 CC | Read about this section in the .text |
| Virtual address (4) | 0x00 00 30 00 | Read about this section in the .text |
| Raw size (SizeOfRawData) (4) | 0x00 00 04 00 | Read about this section in the .text |
| Raw offset (PointerToRawData) (4) | 0x00 00 0A 00 | Read about this section in the .text |
| Pointer To Relocations (4) | 0 | Read about this section in the .text |
| Pointer To Line numbers (4) | 0 | Read about this section in the .text |
| Number Of Relocations (2) | 0 | Read about this section in the .text |
| Number Of Line numbers (2) | 0 | Read about this section in the .text |
| Characteristics (4) | 0xC0 00 00 40 | Read about this section in the .text |

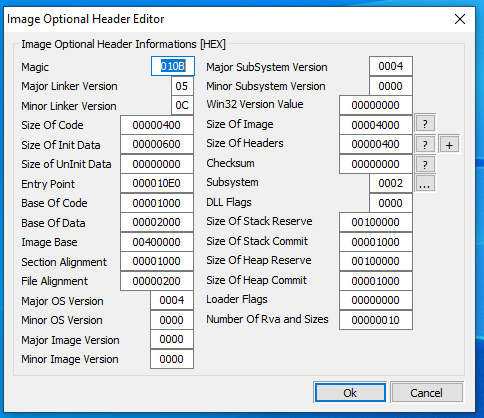
Start Of Pe Header = 0x00 00 00 C0

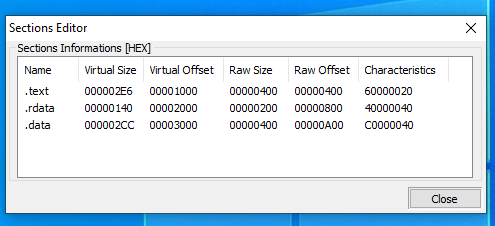
Size Of Optional Header= 0x00E0

Start of Section = Start Of Pe Header + 24 + Size Of Optional Header

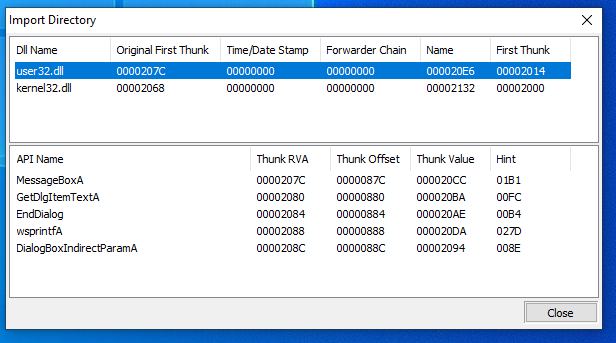


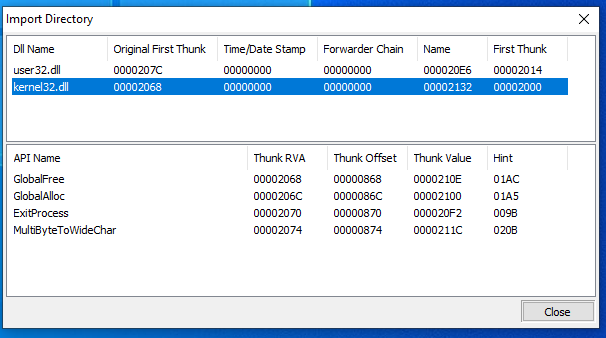






# Import directories

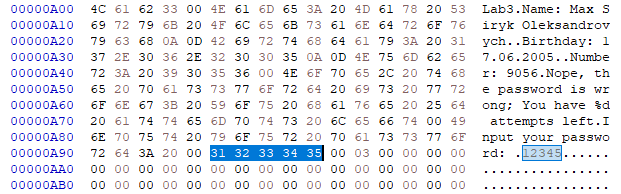




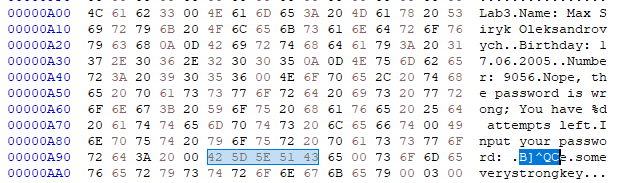
Отже, у програмі використовується 2 бібліотеки user32.dll i kernel32.dll. На скриншотх можна побачити перелік цих функцій.

# Password

Щоб знайти початок потрібно знайти .data, і в ній Raw Offset (0x00 00 0A 00)

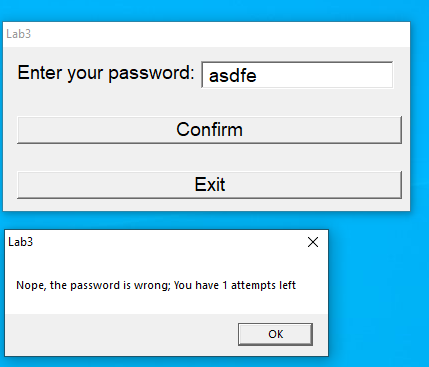


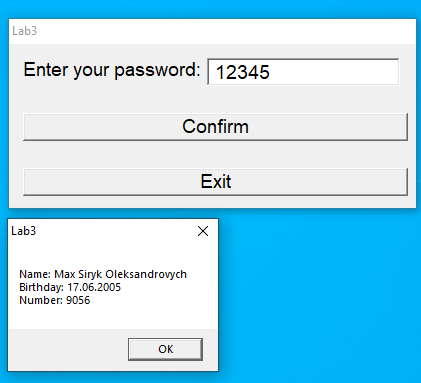
# **“Зашифрований” пароль**



На даному скриншоті представлений як **Hex, так і Text** формати, ліворуч та праворуч відповідно. Ключ “someverystrongkey”.

# Програма з шифруванням





# Висновок

Отже, в ході лабораторної роботи № 3 я ознайомився з форматам PE. Під час виконання лаб. Роб. Дізнався, як зберігаються дані, в тому числі й паролі. У звичайній програмі можна без надзусиль його віднайти, в той час як у програмі з шифруванням ми можемо знайти тільки зашифрований пароль та ключ. Звісно, наше рішення було б “безпечнішим”, якби ми використали сучасні алгоритми шифрування, а не XOR; хоча воно безпечніше ніш у початковій програмі. Також варто зазначити, що шифрування паролю без шифрування “секретної інформації” є безглуздим, адже її також можна віднайти як і пароль.