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 №4**

з дисципліни «Системне програмування» на тему

«Макровизначення і макроси в MASM»

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**: Вивчити технології створення і вживання макросів. Дослідження результатів роботи макрогенератору **MASM**.

**Порядок виконання роботи**

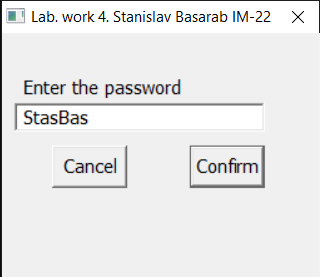
1. Вивчити правила використання макросів в програмах.
2. Переробити програму, підготовлену в лабораторній роботі 3 так, щоб виведення кожного виду інформації (дата народження, номер залікової книжки і т. п.) виконувалося у вигляді багатократного звернення до макросу (макрос 1), в якому у віконному інтерфейсі здійснювалося б виведення тексту, який передається в макрос як параметр.
3. Оформити також у вигляді окремих макросів фрагменти програми, де здійснюється шифрування введеного рядка символів (макрос 2), а також порівняння її з хеш-кодом оригіналу пароля, що зберігається у програмі (макрос 3). Макровизначення для всіх макросів розмістити в тому ж файлі, що і програма. У всіх макросах обов'язково використовувати звичайні і приховані коментарі, а в останньому макросі також використовувати механізм оголошення локальних міток.
4. Виконати компіляцію і компоновку файлу програми.
5. Перевірити роботу програми шляхом введення як правильного, так і невірного паролів.
6. Отримати розширений лістинг програми за допомогою опції **/Fl** компілятора **ML**.
7. Провести дослідження отриманого лістингу: визначити, яким чином компілятор виконав трансляцію кожній з команд макросів і помітити їх в тексті розширеного лістингу кольоровими олівцями або кольоровими фломастерами.
8. Повторити дослідження, які виконані в п. 4 – 7 для другої версії програми, де всі перераховані в п. 2 і 3 макровизначення для макросів повинні розташовуватися в окремому файлі.
9. Порівняти тексти розширених лістингів для обох варіантів розміщення макровизначень, знайти схожість і відмінності. Відобразити їх в звіті по лабораторній роботі.
10. Зробити висновки по лабораторній роботі.

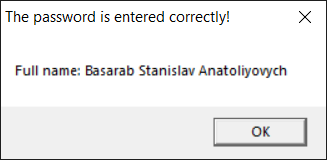
**Виконання роботи**

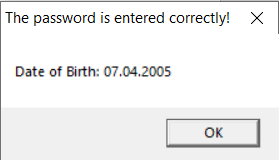
Правильний пароль – StasBas

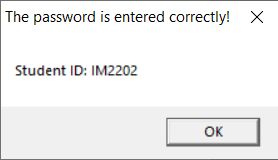
**Дослідження першої програми (усе в одному файлі)**

Під час введення правильного паролю відбувається багатократний виклик макросу для відображення вікон з інформацією про студента.

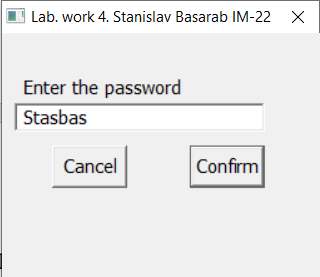


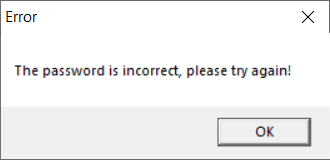






Під час введення неправильного паролю відбувається однократний виклик макросу.





Згідно з завданням було написано **3 макроси**:

* для виводу вікна з певною інформацією (приймає два аргументи)
* макрос для шифрування паролю операцією xor
* макрос для порівняння паролів (приймає два аргументи)

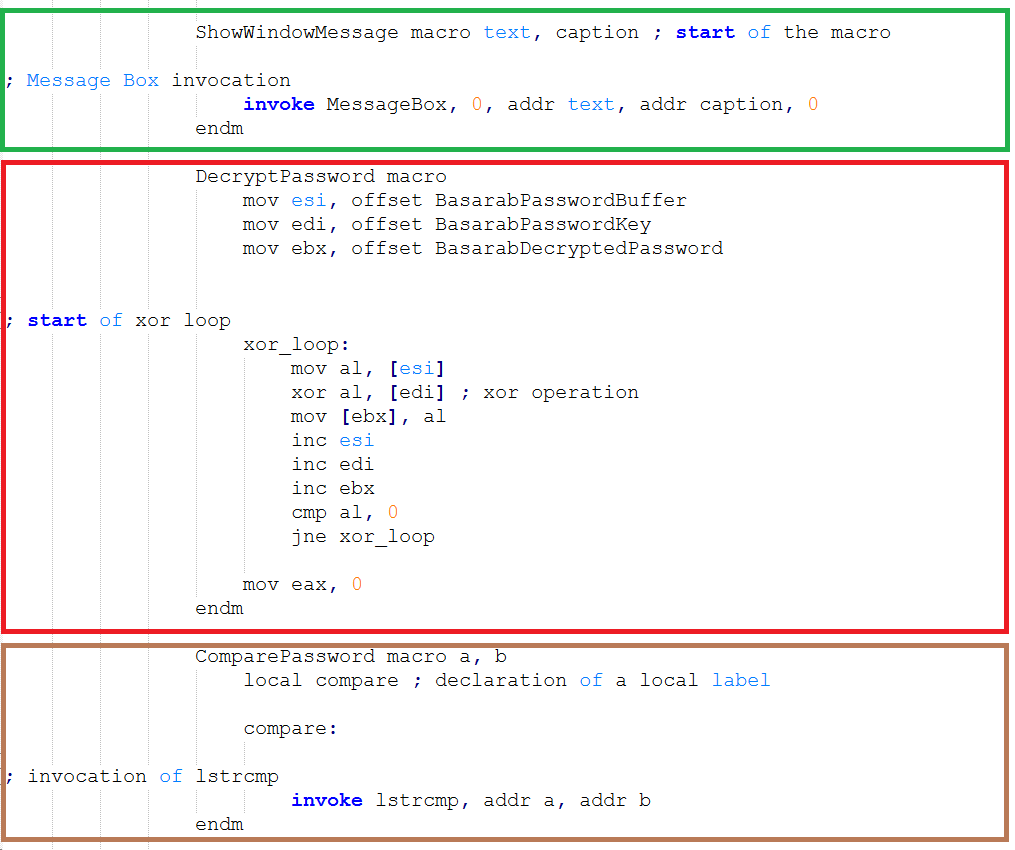
Шифрування пароля за допомогою функції XOR

Зашифруємо пароль за допомогою ключа: «8395027»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ASCII | Binary | | | | | | |
| StasBas | 01010011 | 01110100 | 01100001 | 01110011 | 01000010 | 01100001 | 01110011 |
| 8395027 | 00111000 | 00110011 | 00111001 | 00110101 | 00110000 | 00110010 | 00110111 |
| Результат після операції XOR | | | | | | | |
| kGXFrSD | 01101011 | 01000111 | 01011000 | 01000110 | 01110010 | 01010011 | 01000100 |

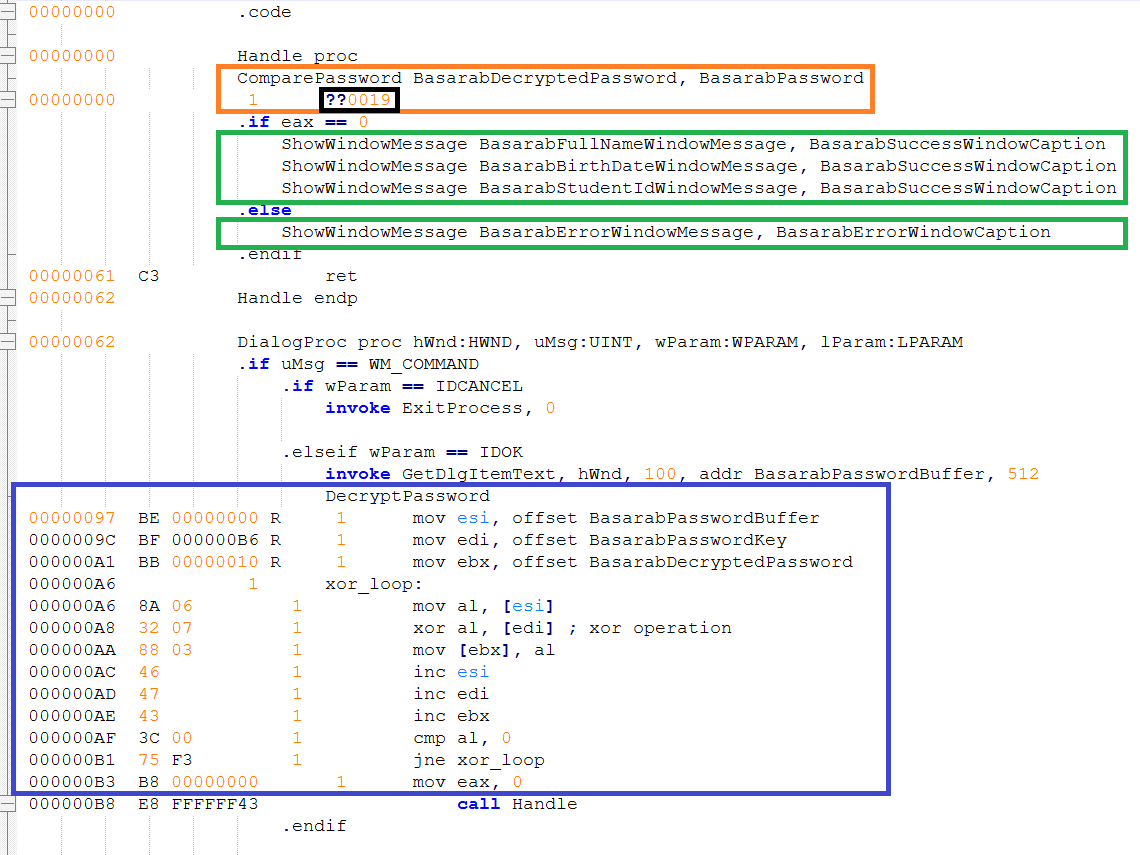
Зашифрований пароль: kGXFrSD

**Дослідження лістингу**  
Макровизначення



Як ми бачимо, на скріншоті зображені та обведені макровизначення для відображення вікна з повідомленням (зеленим кольором), для шифрування введеного рядка символів (червоним кольором), для порівняння цього ж рядка з хеш-кодом оригіналу пароля (коричневим кольором). Також варто зазначити, що ми можемо побачити як звичайні, так і приховані коментарі. А от в останньому макросі використана локальна мітка з оригінальною назвою.

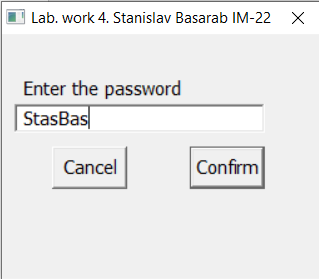
Трансляції макросів

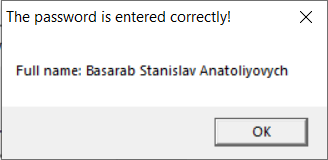


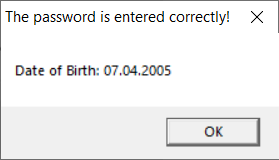
Вище можна побачити трансляцію усіх трьох макросів: для порівняння рядка символів з хеш-кодом оригіналу пароля (помаранчевий колір), для відображення вікна з повідомленням (зеленим кольором), для шифрування введеного рядка символів (синім кольором). Варто зазначити, що у трансляції макросу для порівняння паролів немає оригінальної назви мітки, замість цього є код (обведено чорним кольором). Також після трансляції немає прихованих коментарів (лише звичайні). Важливим моментом є те, що виклики функцій invoke приховані у макросах для порівняння паролів та для відображення вікна з повідомленням.

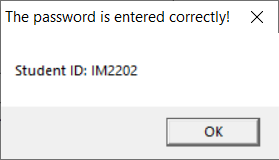
**Дослідження другої програми (макровизначення в окремому файлі)**

Як і в першій програмі під час введення правильного паролю відбувається багатократний виклик макросу для відображення вікон з інформацією про студента.

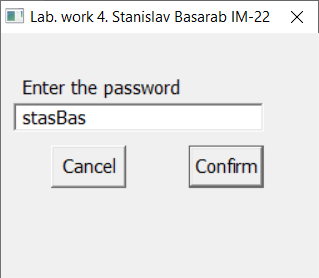


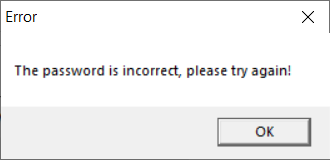






Під час введення неправильного паролю відбувається однократний виклик макросу.



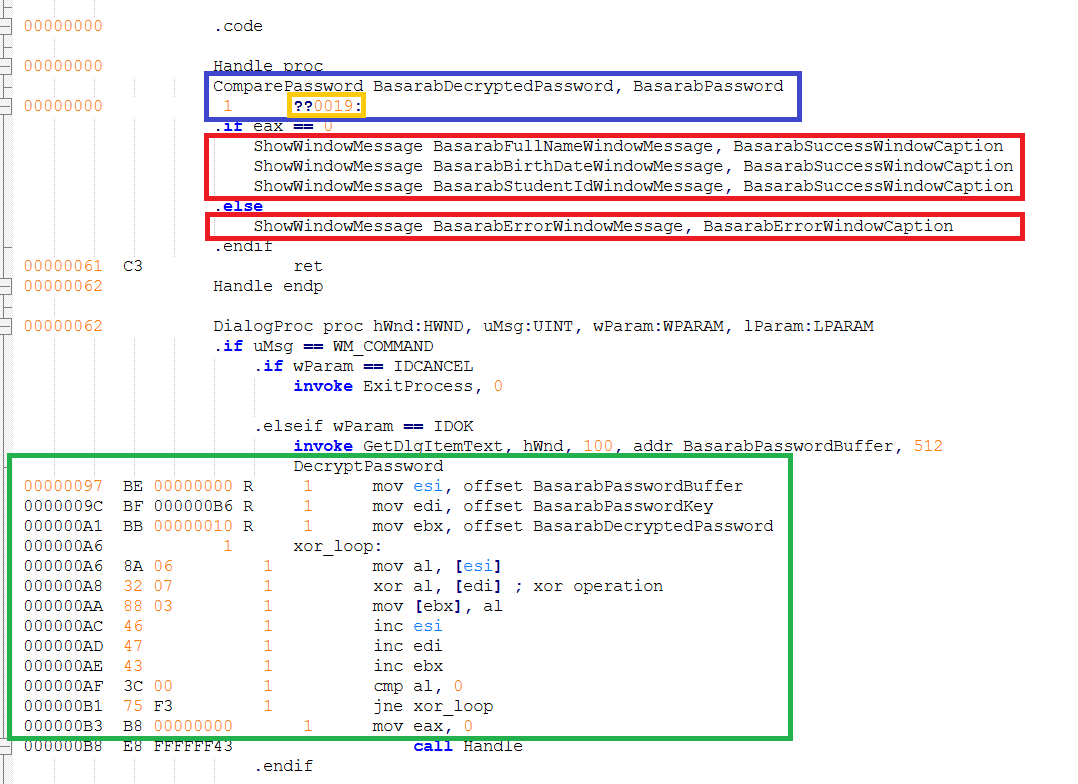


**Дослідження лістингу**  
Макровизначення



Усі три макровизначення у лістингу можна відразу побачити після їх імпорту (обведено синім кольором), а саме: для відображення вікна з повідомленням (червоним кольором), для шифрування введеного рядка символів (жовтим кольором), для порівняння цього ж рядка з хеш-кодом оригіналу пароля (зеленим кольором). Аналогічно до першої програми у макровизначеннях можна побачити як приховані, так і звичайні коментарі. Також в останньому макросі є локальна мітка з оригінальною назвою.

Трансляції макросів



Як і в першій програмі на скріншоті обведено трансляцію усіх трьох макросів: для порівняння рядка символів з хеш-кодом оригіналу пароля (синій колір), для відображення вікна з повідомленням (червоним кольором), для шифрування введеного рядка символів (зеленим кольором). Аналогічно до першої програми у трансляції макросу для порівняння паролів немає оригінальної назви мітки, замість цього є код (обведено жовтим кольором). Також після трансляції немає прихованих коментарів (лише звичайні). Важливим моментом є те, що виклики функцій invoke приховані у макросах для порівняння паролів та для відображення вікна з повідомленням.

**Результати порівняння**

У цілому ми бачимо, що макровизначення та трансляції макросів є досить подібними в обох програмах. Проте є деякі відмінності, як от розташування макровизначень у лістингу файла. На відміну від першої програми, де макровизначення розташовувалися в тому ж місці, де були прописані в коді, у другій програмі вони йдуть одразу після їх імпорту (include 4-2-IM-22-Basarab.inc). А от ситуація з міткою абсолютно однакова. Її оригінальна назва замінюється кодом в обох програмах. Також приховані коментарі не відображаються в трансляціях макросів обох програм. Ну і насамкінець, можна бачити, що виклики функцій invoke приховані в макросах обох програм.

**Висновки**

Отже, під час виконання лабораторної роботи було вивчено технології створення і вживання макросів. Досліджено результати роботи макрогенератору MASM. Написано дві програми, у першій – макровизначення розташовані разом з кодом усієї програми, у другій – макровизначення розташовано в окремому файлі .inc. Досліджено макровизначення та трансляції макросів з лістингів обох програм. За цим дослідженням складені результати порівняння.

Подібне: в макровизначеннях можна побачити як приховані, так і звичайні коментарі, а от у трансляціях макросів є тільки звичайні коментарі, а прихованих немає; також оригінальна назва мітки у макросі 3 замінюється на код в обох програмах; код макросів вставляється відразу після їх виклику в трансляціях; виклики функцій invoke приховані у макросах для порівняння паролів та для відображення вікна з повідомленням.

Відмінне: розташування макровизначень у лістингу програм, через те, що в першій програми макровизначення написані разом з кодом програми, а другій – відбувається їх імпорт з окремого файлу.

Також в обох програмах використані приховані й звичайні коментарі. На основі дослідження можна сказати, що прихованих коментарів не видно в трансляціях макросів, на відміну від звичайних. А от у макровизначеннях можна побачити обидва типи коментарів. Також у третьому макросі ініціалізована за допомогою «local» та використана локальна мітка. У обох програмах у трансляціях макросів вона замінюється на унікальний код.

У цілому макроси дозволяють оптимізувати написання коду та перевикористовувати одні і ті ж елементи коду за допомогою виклику макросу з аргументами чи без.