**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 у MASM32”

Виконав: Перевірив:

студент групи ІМ-22 доцент

Куц Іван Васильович Павлов В.Г.

Номер у списку групи: 12

4 останні цифри студентського: 9017

**Київ 2024**

1. Відрефакторимо код з попередньої лабораторної додавши 3 макроси:
   1. Макрос виклику вікна **callDialogWindow**
   2. Макрос шифрування паролю **encryptPassword**
   3. Макрос порівняння зашифрованого та введеного паролів **comparePassword**

Зробимо компіляцію, лінкування та запустимо програму, спостерігаємо:

**Зображення, що містить текст, знімок екрана, Прямокутник, Шрифт

Автоматично згенерований опис**

Мал. 1. Вікно для введення паролю

**Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис**

Мал. 2. Вікна з персональної інформацією студента, коли пароль введено правильно

Зображення, що містить текст, знімок екрана, Шрифт, число

Автоматично згенерований опис

Мал. 3. Вікно з повідомленням про неправильно введений пароль

1. Розберемо принцип шифрування паролю

У програмі воно відбувається застосуванням операції XOR для певного набору символі з відповідними символами певного ключа

|  |  |
| --- | --- |
| A = 6510 = 010000012  S = 8310 = 010100112  S = 8310 = 010100112  M = 7710 = 010011012  3 = 5110 = 001100112  2 = 5010 = 001100102 | 1 = 4910 = 001100012  2 = 5010 = 001100102  3 = 5110 = 001100112  4 = 5210 = 001101002  q = 11310 = 011100012  w = 11910 = 011101112 |

Процес шифрування:

A XOR 1 = p

S XOR 2 = a

S XOR 3 = `

M XOR 4 = y

3 XOR q = B

2 XOR w = E

Отже, маємо зашифрований пароль - **pa`yBE**

1. Винесемо макроси у окремий файл, назвавши його 4-12-IM-22-Kuts.inc та імпортуємо їх з цього файлу у новий файл програми 4-12-IM-22-Kuts-INC.asm
2. Зробимо лістинги обох файлів програми за допомогою опції /Fl компілятора ML.
3. Проаналізуємо визначення макросів у лістингу першої програмі

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Мал. 4. Визначення макросів у лістингу 4-12-IM-22-Kuts.asm файлу

На малюнку вище бачимо визначення макросів **callDialogWindow, encryptPassword** та **comparePassword**

Можемо спостерігати, що приховані коментарі поки що видно, також ми використали локальну мітку та відмітили її помаранчевим на малюнку, пізніше побачимо її поведінку під час трансляції

1. Проаналізуємо трансляцію макросів у лістингу першої програмі

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Мал. 5.Трансляція макросів у лістингу 4-12-IM-22-Kuts.asm файлу

На малюнку вище бачимо трансляцію макросів **callDialogWindow, encryptPassword** та **comparePassword**

Варто відмітити, що в порівнянні з визначенням макросів на попередньому малюнку ми не бачимо коментарів.

Макрос **encryptPassword** та **comparePassword** вставляються у програмний код, єдине що у макросі **comparePassword,** де було використано локальну мітку – вона закодована (виділено помаранчевим кольором)

Також, можемо бачити що не відбулось трансляції макроса **callDialogWindow ,** він лишився у вигляді маркроса

1. Проаналізуємо визначення макросів у лістингу другої програмі

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Шрифт

Автоматично згенерований опис**

Мал. 6. Визначення макросів у лістингу 4-12-IM-22-Kuts.asm файлу

На малюнку вище бачимо визначення макросів **callDialogWindow, encryptPassword** та **comparePassword** у лістингу програми 4-12-IM-22-Kuts-INC.asm

Дані макроси були вставлені у програмний код одразу після директиви include

Як бачимо, код не відрізняється від визначення у першому файлі, коментарі також видно, локальні мітки не шифруються

1. Проаналізуємо трансляцію макросів у лістингу другої програми

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис

Мал. 7.Трансляція макросів у лістингу 4-12-IM-22-Kuts-INC.asm файлу

На малюнку вище бачимо трансляцію макросів **callDialogWindow, encryptPassword** та **comparePassword**

Як і у варіанті першої програми можемо спостерігати що ми не бачимо прихованих коментарів.

Макроси **encryptPassword** та **comparePassword** так само вставляються у програмний код. У макросі **comparePassword,** де було використано локальну мітку – вона закодована (виділено помаранчевим кольором)

Так само як і в першому варіанті, можемо бачити що не відбулось трансляції макроса **callDialogWindow ,** він лишився у вигляді маркроса

**Висновок**

Під час виконання лабораторної роботи було досліджено та використано на практиці макроси у мові MASM32, було досліджено їх поведінку при різних видах застосування (коли визначається разом з програмним кодом та коли імпортуються з іншого файлу). Різниці в 2 випадках використання макросів небагато, єдина відмінність у тому, що у 2 варіанті макроси вставляються на місці директиви **include**

Спільного в 2 варіантах використання чимало:

* Видно як звичайні, так і приховані коментарі
* При трансляції приховані коментарі більше не видно;
* Назви локальних міток змінюються на унікальні коди
* Приховано виклик функції з використанням команди invoke. Це добре видно у 1 макросі, коли в файлі трансляції ми бачимо лише виклик макрос з аргументам и
* Код макросів вставляється під їх викликом

Варто відмітити користь макросів. Вони дозволяють створювати компактні та зрозумілі блоки коду, що можуть бути легко використані у різних частинах програми. Такий підхід значно спрощує розробку програмного забезпечення та полегшує підтримку коду в подальшому.