Мiнiстерство освiти i науки України  
Нацiональний технiчний унiверситет України «КПI» iм. I. Сiкорського

Факультет iнформатики та обчислювальної технiки

Кафедра обчислювальної технiки

**ЗВІТ**

до лабораторної роботи № 4

на тему:

**«Макровизначення і макроси в MASM32»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав**  **студент** |  | *ІП-64 Вінницький В'ячеслав Андрієвич* |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Прийняв** |  | *Павлов В.Г.* |  |  |
|  |  |  |  |  |

Київ 2017

# Постановка Задачі

## Мета

Вивчити технології створення і вживання макросів. Дослідження результатів роботи макрогенератору **MASM32**.

## Порядок виконання

1. Вивчити правила використання макросів в програмах [1].

2. Переробити програму, підготовлену в лабораторній роботі 6 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.

# Лістинги програмного коду

## Із заданням макросів в тому ж файлі:

.386 ; Enable 80386+ instruction set

.model flat, stdcall

option casemap: none

include c:\masm32\include\masm32rt.inc

.data

promt db "Enter password: ", 0

chars\_read dd ?

chars\_written dd ?

entered\_password db 20 dup(?)

password db '1234'

error\_mes db "Error, password incorrect", 0

st\_data\_name db "Vinnitskyi Viacheslav Andrievich", 0

st\_data\_date db "Birhday : 30.12.1998",0

st\_data\_student\_book db "Number - 6402", 0

title\_data db "Info", 0

title\_error db "Error", 0

.code

OutputInfo macro str\_d, str\_t ; макрос данных вывода

invoke MessageBox, 0, str\_d, str\_t, 0 ;; макрос данных вывода

endm

MacroXor macro str\_xor, num

local cmp\_m

xor ecx, ecx ; ecx = 0

cmp\_m:

mov dl, entered\_password[ecx] ;; Строка не будет видна после подстановки макроса

xor dl, num ; сравнение

mov entered\_password[ecx], dl

inc ecx

cmp ecx, lengthof password

jne cmp\_m

endm

MacroCMP macro str1, str2

local string\_cmp ;

xor ecx, ecx

string\_cmp:

mov dl, str1[ecx] ; load dl with char from str1

mov dh, str2[ecx] ; load dh with char from str2

cmp dl, dh ;; Строка не будет видна после подстановки макроса

jne errorer ;;

inc ecx

cmp ecx, lengthof password

jne string\_cmp

endm

start:

invoke AllocConsole

invoke SetConsoleOutputCP, 1251

invoke GetStdHandle, STD\_INPUT\_HANDLE

mov ebx, eax

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

invoke WriteConsoleA, eax, addr promt, sizeof promt, addr chars\_written, 0

invoke ReadConsoleA, ebx, addr entered\_password, sizeof entered\_password, addr chars\_read, 0

MacroXor entered\_password, 5

MacroCMP entered\_password, password

OutputInfo addr st\_data\_name, addr title\_data

OutputInfo addr st\_data\_date, addr title\_data

OutputInfo addr st\_data\_student\_book, addr title\_data

jmp @end

errorer:

OutputInfo addr error\_mes, addr title\_error

@end:

invoke ExitProcess, 0

end start

## Із заданням макросів в окремому файлі

### Файл .inc:

OutputInfo macro str\_d, str\_t ; macro of data output

invoke MessageBox, 0, str\_d, str\_t, 0 ;; macro of data output

endm

MacroXor macro str\_xor, num

local cmp\_m

xor ecx, ecx ; ecx = 0

cmp\_m:

mov dl, entered\_password[ecx] ;; you wont see this line after substitution

xor dl, num ; dl ^ num

mov entered\_password[ecx], dl

inc ecx

cmp ecx, lengthof password

jne cmp\_m

endm

MacroCMP macro str1, str2

local string\_cmp ; some local mark

xor ecx, ecx

string\_cmp:

mov dl, str1[ecx] ; load dl with char from str1

mov dh, str2[ecx] ; load dh with char from str2

cmp dl, dh ;; you wont see this line after substitution

jne errorer ;; jump

inc ecx

cmp ecx, lengthof password

jne string\_cmp

endm

### Файл .asm:

.386 ; Enable 80386+ instruction set

.model flat, stdcall

option casemap: none

include c:\masm32\include\masm32rt.inc

include lab4inc.inc

.data

promt db "Enter password: ", 0

chars\_read dd ?

chars\_written dd ?

entered\_password db 20 dup(?)

password db '1234'

error\_mes db "Error, password incorrect", 0

st\_data\_name db "Vinnitskyi Viacheslav Andrievich", 0

st\_data\_date db "Birthday : 30.12.1998",0

st\_data\_student\_book db "Number in grade book - 6402", 0

title\_data db "Info", 0

title\_error db "Error", 0

.code

start:

invoke AllocConsole

invoke SetConsoleOutputCP, 1251

invoke GetStdHandle, STD\_INPUT\_HANDLE

mov ebx, eax

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

invoke WriteConsoleA, eax, addr promt, sizeof promt, addr chars\_written, 0

invoke ReadConsoleA, ebx, addr entered\_password, sizeof entered\_password, addr chars\_read, 0

MacroXor entered\_password, 5

MacroCMP entered\_password, password

OutputInfo addr st\_data\_name, addr title\_data

OutputInfo addr st\_data\_date, addr title\_data

OutputInfo addr st\_data\_student\_book, addr title\_data

jmp @end

errorer:

OutputInfo addr error\_mes, addr title\_error

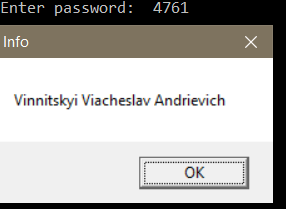
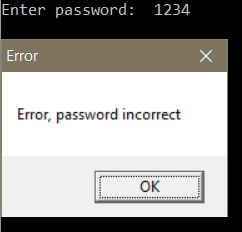
@end:

invoke ExitProcess, 0

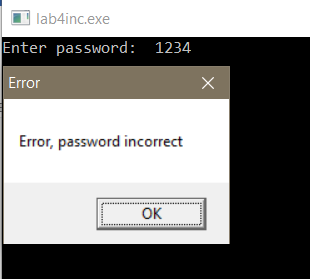
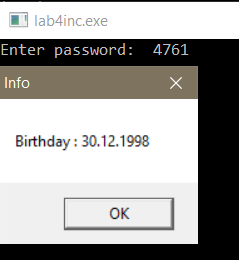
end start

## Результат роботи програми

### Із заданням макросів в тому ж файлі

### Із заданням макросів в окремому файлі

# РозШирені Лістинги

## Із заданням макросів в тому ж файлі

00000000 .data

00000000 45 6E 74 65 72 promt db "Enter password: ", 0

20 70 61 73

73 77 6F 72

64 3A 20 00

00000011 00000000 chars\_read dd ?

00000015 00000000 chars\_written dd ?

00000019 00000014 [ entered\_password db 20 dup(?)

00

]

0000002D 31 32 33 34 password db '1234'

00000031 45 72 72 6F 72 error\_mes db "Error, password incorrect", 0

2C 20 70 61

73 73 77 6F

72 64 20 69

6E 63 6F 72

72 65 63 74

00

0000004B 56 69 6E 6E 69 st\_data\_name db "Vinnitskyi Viacheslav Andrievich", 0

74 73 6B 79

69 20 56 69

61 63 68 65

73 6C 61 76

20 41 6E 64

72 69 65 76

69 63 68 00

0000006C 42 69 72 68 64 st\_data\_date db "Birhday : 30.12.1998",0

61 79 20 3A

20 33 30 2E

31 32 2E 31

39 39 38 00

00000081 4E 75 6D 62 65 st\_data\_student\_book db "Number - 6402", 0

72 20 2D 20

36 34 30 32

00

0000008F 49 6E 66 6F 00 title\_data db "Info", 0

00000094 45 72 72 6F 72 title\_error db "Error", 0

00

00000000 .code

OutputInfo macro str\_d, str\_t ; ьръЁюё фрээ√ї т√тюфр

invoke MessageBox, 0, str\_d, str\_t, 0 ;; ьръЁюё фрээ√ї т√тюфр

endm

MacroXor macro str\_xor, num

local cmp\_m

xor ecx, ecx ; ecx = 0

cmp\_m:

mov dl, entered\_password[ecx] ; ╤ЄЁюър эх сєфхЄ тшфэр яюёых яюфёЄрэютъш ьръЁюёр

xor dl, num ; ёЁртэхэшх

mov entered\_password[ecx], dl

inc ecx

cmp ecx, lengthof password

jne cmp\_m

endm

MacroCMP macro str1, str2

local string\_cmp ;

xor ecx, ecx

string\_cmp:

mov dl, str1[ecx] ; load dl with char from str1

mov dh, str2[ecx] ; load dh with char from str2

cmp dl, dh ; ╤ЄЁюър эх сєфхЄ тшфэр яюёых яюфёЄрэютъш ьръЁюёр

jne errorer ;;

inc ecx

cmp ecx, lengthof password

jne string\_cmp

endm

00000000 start:

invoke AllocConsole

invoke SetConsoleOutputCP, 1251

invoke GetStdHandle, STD\_INPUT\_HANDLE

00000016 8B D8 mov ebx, eax

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

invoke WriteConsoleA, eax, addr promt, sizeof promt, addr chars\_written, 0

invoke ReadConsoleA, ebx, addr entered\_password, sizeof entered\_password, addr chars\_read, 0

MacroXor entered\_password, 5

00000047 33 C9 1 xor ecx, ecx ; ecx = 0

00000049 1 ??0019:

00000049 8A 91 00000019 R 1 mov dl, entered\_password[ecx]

0000004F 80 F2 05 1 xor dl, 5 ; ёЁртэхэшх

00000052 88 91 00000019 R 1 mov entered\_password[ecx], dl

00000058 41 1 inc ecx

00000059 83 F9 04 1 cmp ecx, lengthof password

0000005C 75 EB 1 jne ??0019

MacroCMP entered\_password, password

0000005E 33 C9 1 xor ecx, ecx

00000060 1 ??001A:

00000060 8A 91 00000019 R 1 mov dl, entered\_password[ecx] ; load dl with char from str1

00000066 8A B1 0000002D R 1 mov dh, password[ecx] ; load dh with char from str2

0000006C 38 F2 1 cmp dl, dh

0000006E 75 41 1 jne errorer

00000070 41 1 inc ecx

00000071 83 F9 04 1 cmp ecx, lengthof password

00000074 75 EA 1 jne ??001A

OutputInfo addr st\_data\_name, addr title\_data

OutputInfo addr st\_data\_date, addr title\_data

OutputInfo addr st\_data\_student\_book, addr title\_data

000000AF EB 13 jmp @end

000000B1 errorer:

OutputInfo addr error\_mes, addr title\_error

000000C4 @end:

invoke ExitProcess, 0

end start

## Із заданням макросів в окремому файлі

include lab4inc.inc

C OutputInfo macro str\_d, str\_t ; macro of data output

C invoke MessageBox, 0, str\_d, str\_t, 0 ;; macro of data output

C endm

C MacroXor macro str\_xor, num

C local cmp\_m

C

C xor ecx, ecx ; ecx = 0

C cmp\_m:

C mov dl, entered\_password[ecx] ;; you wont see this line after substitution

C xor dl, num ; dl ^ num

C mov entered\_password[ecx], dl

C inc ecx

C cmp ecx, lengthof password

C jne cmp\_m

C endm

C

C MacroCMP macro str1, str2

C local string\_cmp ; some local mark

C xor ecx, ecx

C string\_cmp:

C mov dl, str1[ecx] ; load dl with char from str1

C mov dh, str2[ecx] ; load dh with char from str2

C cmp dl, dh ; you wont see this line after substitution

C jne errorer ;; jump

C inc ecx

C cmp ecx, lengthof password

C jne string\_cmp

C endm

C

00000000 .data

00000000 45 6E 74 65 72 promt db "Enter password: ", 0

20 70 61 73

73 77 6F 72

64 3A 20 00

00000011 00000000 chars\_read dd ?

00000015 00000000 chars\_written dd ?

00000019 00000014 [ entered\_password db 20 dup(?)

00

]

0000002D 31 32 33 34 password db '1234'

00000031 45 72 72 6F 72 error\_mes db "Error, password incorrect", 0

2C 20 70 61

73 73 77 6F

72 64 20 69

6E 63 6F 72

72 65 63 74

00

0000004B 56 69 6E 6E 69 st\_data\_name db "Vinnitskyi Viacheslav Andrievich", 0

74 73 6B 79

69 20 56 69

61 63 68 65

73 6C 61 76

20 41 6E 64

72 69 65 76

69 63 68 00

0000006C 42 69 72 74 68 st\_data\_date db "Birthday : 30.12.1998",0

64 61 79 20

3A 20 33 30

2E 31 32 2E

31 39 39 38

00

00000082 4E 75 6D 62 65 st\_data\_student\_book db "Number in grade book - 6402", 0

72 20 69 6E

20 67 72 61

64 65 20 62

6F 6F 6B 20

2D 20 36 34

30 32 00

0000009E 49 6E 66 6F 00 title\_data db "Info", 0

000000A3 45 72 72 6F 72 title\_error db "Error", 0

00

00000000 .code

00000000 start:

invoke AllocConsole

invoke SetConsoleOutputCP, 1251

invoke GetStdHandle, STD\_INPUT\_HANDLE

00000016 8B D8 mov ebx, eax

invoke GetStdHandle, STD\_OUTPUT\_HANDLE

invoke WriteConsoleA, eax, addr promt, sizeof promt, addr chars\_written, 0

invoke ReadConsoleA, ebx, addr entered\_password, sizeof entered\_password, addr chars\_read, 0

MacroXor entered\_password, 5

00000047 33 C9 1 xor ecx, ecx ; ecx = 0

00000049 1 ??0019:

00000049 8A 91 00000019 R 1 mov dl, entered\_password[ecx]

0000004F 80 F2 05 1 xor dl, 5 ; dl ^ num

00000052 88 91 00000019 R 1 mov entered\_password[ecx], dl

00000058 41 1 inc ecx

00000059 83 F9 04 1 cmp ecx, lengthof password

0000005C 75 EB 1 jne ??0019

MacroCMP entered\_password, password

0000005E 33 C9 1 xor ecx, ecx

00000060 1 ??001A:

00000060 8A 91 00000019 R 1 mov dl, entered\_password[ecx] ; load dl with char from str1

00000066 8A B1 0000002D R 1 mov dh, password[ecx] ; load dh with char from str2

0000006C 38 F2 1 cmp dl, dh

0000006E 75 41 1 jne errorer

00000070 41 1 inc ecx

00000071 83 F9 04 1 cmp ecx, lengthof password

00000074 75 EA 1 jne ??001A

OutputInfo addr st\_data\_name, addr title\_data

OutputInfo addr st\_data\_date, addr title\_data

OutputInfo addr st\_data\_student\_book, addr title\_data

000000AF EB 13 jmp @end

000000B1 errorer:

OutputInfo addr error\_mes, addr title\_error

000000C4 @end:

invoke ExitProcess, 0

end start

ВИСНОВКИ

У ході виконання лабораторної роботи було здобуті навички розробки та використання макросів та їх використання. Результатом даної лабораторної роботи є програма, яка при введенні правильного паролю виводить на екран данні про студента. Також, було сформовано розширені лістинги програми з макросами в тому ж і в окремому файлі.

Аналізуючи лістинги, можна помітити що після підстановки макросів програмний код ніяк не відрізняється в обох версіях. Якщо макровизначення написати в тому ж файлі то вони залишаться в тому ж місці де вони були визначенні, але якщо перемістити їх в окремий .inc файл, то вони вставляться в програмний код в те міце де була написана команда include.