

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Лабораторна робота №7

З дисципліни: “ Системне програмування ”

Тема: "Створення DLL та їх використання при явному зв'язуванні на мові Асемблер "

Варіант 12

Виконав: ст. гр. КІ-38

Папіш Олександр Ростиславович

Керівник: асистент каф. ЕОМ

Козак Н.Б.

ЛЬВІВ 2020

Мета роботи

Ознайомитись з технологією та оволодіти навиками створення та використання бібліотек динамічного компонування з використанням явного зв'язування на мові Асемблера

Завдання

12. Надрукувати всі слова тексту, відмінні від останнього слова.

Код програми

```
// lab7dll.def
```

```
LIBRARY lab7dll
EXPORTS
    PrintUniqueWords
```

```
// lab7dll.asm
```

```
.586
.model flat, STDCALL
option casemap: none
include C:\masm32\include\windows.inc
include C:\masm32\macros\macros.asm
include C:\masm32\include\masm32.inc
include C:\masm32\include\gdi32.inc
include C:\masm32\include\user32.inc
include C:\masm32\include\kernel32.inc
includelib C:\masm32\lib\masm32.lib
includelib C:\masm32\lib\gdi32.lib
includelib C:\masm32\lib\user32.lib
includelib C:\masm32\lib\kernel32.lib

.data

lastWord db 100 dup(0)
wordLength dd 0

resultStr db 1000 dup(0)
resultStrLength dd 0

stringIndex dd 0
someword db 100 dup(0)
somewordLength dd 0
messageBoxTitle db "Lab7", 0

isEq dd 0

intJ dd 0
intI dd 0
condition_1 dd 0
condition_2 dd 0

.code
DllEntry PROC hInstDLL:DWORD, reason:DWORD, reserved:DWORD
```

```

    mov eax, 1
    ret
DllEntry ENDP

```

```

PrintUniqueWords PROC startString:DWORD, stringLength:DWORD

```

```

    mov esi, 0 ; i
    mov edi, 0 ; c
    mov eax, startString
    mov ebx, offset lastWord
    xor ecx, ecx
Mark_2:
    mov cl, byte ptr [eax+esi]
    cmp cl, 32
    je Mark_1
    mov cl, byte ptr [eax+esi]
    cmp cl, 9
    je Mark_1
    mov cl, byte ptr [eax+esi]
    cmp cl, 10
    je Mark_1

    mov cl, byte ptr [eax+esi]
    mov edx, wordLength
    mov [ebx+edx], cl

    mov edx, wordLength
    inc edx
    mov wordLength, edx
    inc esi
    cmp esi, stringLength
    jb Mark_2
    jmp Mark_3

Mark_1:
    mov [lastWord+edi], 0
    inc edi
    cmp edi, wordLength
    jb Mark_1
    mov wordLength, 0
    inc esi
    cmp esi, stringLength
    jb Mark_2

Mark_3:

    mov eax, wordLength
    dec eax
    mov wordLength, eax

    mov eax, stringLength
    sub eax, wordLength
    ;dec eax
    mov condition_1, eax
    add eax, 10
    mov condition_2, eax

Mark_4:
    mov someWordLength, 0

    mov edi, offset someWord
    xor eax, eax
    xor ebx, ebx
Mark_5:
    mov [edi+eax], bl
    inc eax
    cmp eax, condition_2
    jb Mark_5

    mov edi, startString

```

```

mov esi, offset someWord
xor eax, eax
Mark_6:
    mov ecx, stringIndex
    mov ebx, someWordLength
    mov eax, 32
    cmp [edi+ecx], al
    je Mark_7
    mov eax, 9
    cmp [edi+ecx], al
    je Mark_7
    mov eax, 10
    cmp [edi+ecx], al
    je Mark_7
    mov dl, [edi+ecx]
    mov [esi+ebx], dl

    mov dl, 0
    cmp [edi+ecx], dl
    je Mark_9
    inc someWordLength
    jmp Mark_9

Mark_7:
    inc stringIndex
    jmp Mark_8

Mark_9:
    inc stringIndex
    inc ecx
    cmp ecx, stringLength
    jb Mark_6
Mark_8:
    mov isEq, 1
    mov eax, someWordLength
    cmp eax, wordLength
    jne Mark_10
    mov edi, offset someWord
    mov esi, offset lastWord
    mov intJ, 0
    xor eax, eax
    Mark_12:
        mov bl, [edi+eax]
        cmp bl, [esi+eax]
        je Mark_13
        mov isEq, 0
        jmp Mark_11

Mark_13:
    inc eax
    cmp eax, someWordLength
    jb Mark_12
    jmp Mark_11
Mark_10:
    mov isEq, 0

Mark_11:
    xor eax, eax
    cmp eax, isEq
    jne Mark_14
    xor eax, eax
    mov edi, offset resultStr
    mov esi, offset someWord
    mov ebx, intI
    Mark_15:
        mov dl, [esi+eax]
        mov [edi+ebx], dl
        inc eax
        inc intJ
        inc ebx
        inc intI

```

```

        cmp eax, someWordLength
        jb Mark_15
        mov dl, 32
        mov [edi+ebx], dl
        inc ebx
        inc intI

Mark_14:
        mov eax, intI
        cmp eax, condition_1
        jb Mark_4

        mov eax, offset resultStr
        push 40h
        push offset MessageBoxTitle
        push offset resultStr
        push 0
        call MessageBoxA
        ret
PrintUniqueWords ENDP

End DllEntry

```

// lab7test.asm

```

.586
.model flat, stdcall
option casemap:none

include C:\masm32\include\windows.inc
include C:\masm32\macros\macros.asm
include C:\masm32\include\masm32.inc
include C:\masm32\include\gdi32.inc
include C:\masm32\include\user32.inc
include C:\masm32\include\kernel32.inc
includelib C:\masm32\lib\masm32.lib
includelib C:\masm32\lib\gdi32.lib
includelib C:\masm32\lib\user32.lib
includelib C:\masm32\lib\kernel32.lib

.data

LibName db "lab7dll.dll",0
FunctionName db "PrintUniqueWords",0
DllNotFound db "Cannot load library",0
AppName db "Load explicit Library", 0
NotFound db "PrintUniqueWords function not found",0

string db "test anon ano some text text test test", 0,0
lengthStr dd 39

.data?
hLib dd ?
PrintUniqueWordsAddr dd ?
.code

start:
    invoke LoadLibraryA, addr LibName
    .if eax == 0
        invoke MessageBoxA, 0, addr DllNotFound, addr AppName, 40h
    .else

```

```

mov hLib, eax
invoke GetProcAddress, hLib, addr FunctionName
.if eax == 0
    invoke MessageBoxA, 0, addr NotFound, addr AppName, 40h
.else
    push lengthStr
    push offset string

    mov PrintUniqueWordsAddr, eax
    call [PrintUniqueWordsAddr]
.endif
invoke FreeLibrary, hLib
.endif
invoke ExitProcess, 0
end start

```

// makeLab7.bat

@echo off

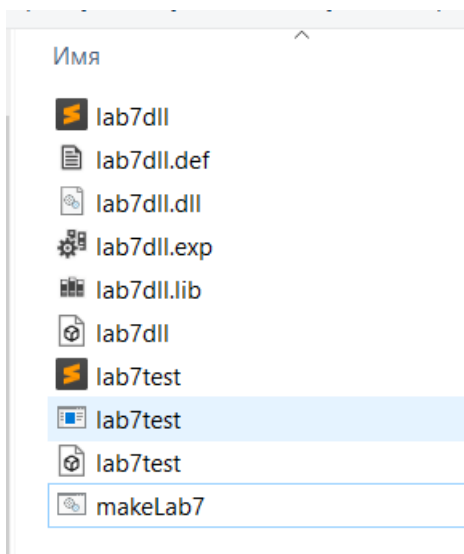
```

\masm32\bin\ml /c /coff lab7dll.asm
\masm32\bin\Link /SUBSYSTEM:WINDOWS /DLL /DEF:lab7dll.def lab7dll.obj

\masm32\bin\ml /c /Zd /coff lab7test.asm
\masm32\bin\Link /SUBSYSTEM:CONSOLE lab7test.obj

```

Створені файли



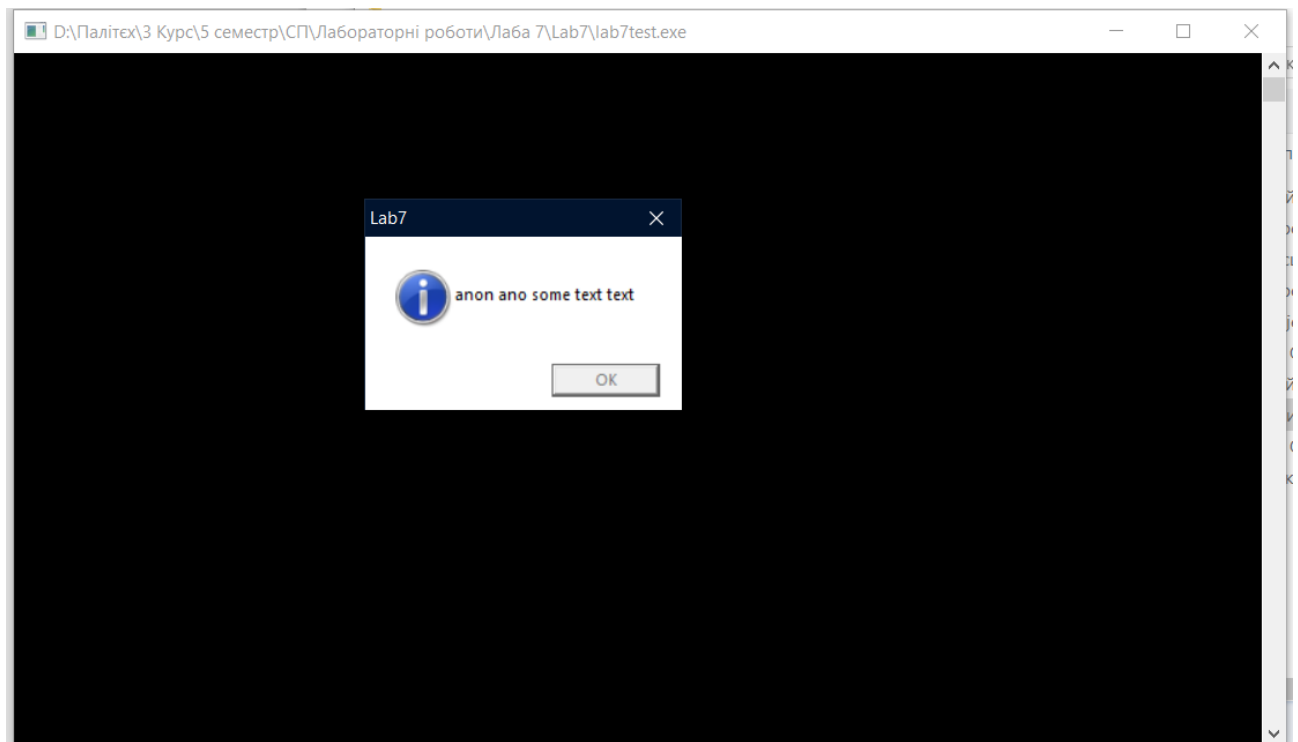
Вхідні дані

```

string db "test anon ano some text text test test", 0
lengthStr dd 39

```

Результат виконання програми



Висновок: на даній лабораторній роботі я ознайомитись з технологією та оволодів навиками створення та використання бібліотек динамічного компонування з використанням явного зв'язування на мові Асемблера