

### Лабораторна робота №5

Тема: «Використання WinAPI для створення програм на мові Assembler»

Мета роботи: навчитися використовуючи команди WinAPI для створення найпростіших програм.

#### Хід роботи

#### **Завдання на 3 бали**

1. Відкомпілювати та відлагодити приклади представлені у роботі. Пояснити їх роботу, стани реєстрів та принципи роботи команд.

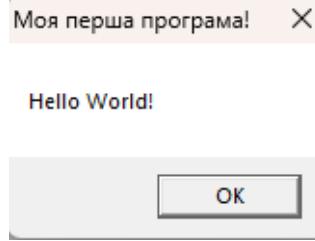
#### **Приклад 1**

```
format PE GUI 4.0
entry start
    include 'win32a.inc'
section '.data' data readable writable
    message db 'Hello World!',0
    title db 'Моя перша програма!',0
    section '.code' code readable executable
start:
push MB_OK
push title
push message
push 0
call [MessageBoxA]
close:
push 0
call [ExitProcess]

section '.idata' import data readable writeable
library kernel,'KERNEL32.DLL',\
user,'USER32.DLL'
import kernel,\
ExitProcess,'ExitProcess'
import user,\
MessageBoxA,'MessageBoxA'
```

#### **Приклад 2**

```
include 'include/win32ax.inc'
.code
main:
    invoke MessageBox,NULL,'Hello, World','Fasm message
box:',MB_OK
    invoke ExitProcess,0
.end main
```



Моя перша програма! X

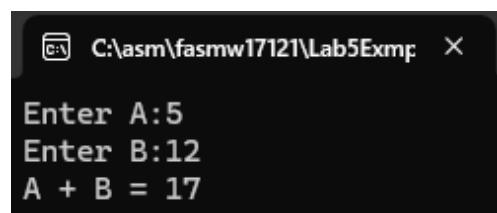
Hello World!

OK

Робота програми

### Приклад 3

```
format PE console 4.0
include 'win32a.inc'
start: cinvoke printf, req, 41h      ; Виведення на екран
       cinvoke scanf, tpt, A          ; введення А
       cinvoke printf, req, 42h      ; Виведення на екран
       cinvoke scanf, tpt, B          ; введення В
       mov eax, [A]
       add eax, [B]
       cinvoke printf, tpo, eax
       invoke sleep, 5000            ; затримка 5 секунд
gtfo:
       invoke exit, 0
req db 'Enter %c:',0
tpo db 'A + B = %d',13,10,0
tpt db '%d',0
A dd ?
B dd ?
data import
library msvcrt,'MSVCRT.DLL', \
           kernel32,'KERNEL32.DLL'
import kernel32, \
           sleep,'Sleep'
import mservcrt, \
           puts,'puts', \
           scanf,'scanf', \
           printf,'printf', \
           exit,'exit'
end data
```



C:\asm\fasmw17121\Lab5Exmp X

Enter A:5  
Enter B:12  
A + B = 17

Робота програми

**Приклад 3(2)**

```

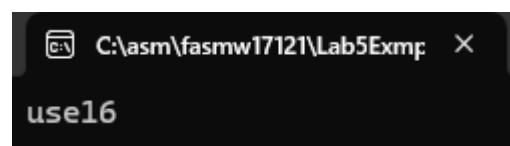
format PE console 4.0
include 'win32a.inc'
start:
    cinvoke fopen,filename,type ;виклик С-функції відкриття
файла
        mov ebx,eax ;зберігає файловий дескриптор
    cinvoke fgets,buf,10,ebx ;виклик С-функції зчитування
файла до 9 символів
        cinvoke printf,buf ;виклик С-функції виводу у консоль
lp2:invoke getch ; 5 sec. delay
gtfo: invoke exit, 0
type db 'r',0 ;режим читання
filename db 'main.asm',0 ;назва файлу
buf db ? ;буфер для зчитаного рядка

;імпорт необхідних бібліотек

data import
library msvcrt,'MSVCRT.DLL',\ ;бібліотека С (файли, printf,
scanf, exit)
    kernel32,'KERNEL32.DLL',\ ;функції Windows (Sleep,
ExitProcess)
    user32,'USER32.DLL' ;робота з мишею, курсором, вікнами

;набір стандартних С-функцій (І/О, робота з файлами,
введення-виведення, завершення програми)
import msvcrt,\
    puts,'puts',\
    scanf,'scanf',\
    printf,'printf',\
    fopen,'fopen',\
    fgets,'fgets',\
    fclose,'fclose',\
    fseek,'fseek',\
    getch,'_getch',\
exit,'exit'
end data

```



Робота програми

## Приклад 5

```

format pe console
entry start
include 'win32ax.inc'
section '.code' code readable executable
start:
    cinvoke system,"Dir"      ; виконує команду cmd – показує
список файлів
    cinvoke system,"Pause"   ; чекає натискання клавіші
    invoke ExitProcess,0     ; завершує програму
section '.idata' import data readable
library kernel32,'KERNEL32.DLL', \
        msvcrt,'MSVCRT.DLL'   ; підключає бібліотеки
import kernel32, \
        ExitProcess,'ExitProcess' ; імпорт функції
завершення
import mservcrt, \
        system,'system'; імпорт функції виконання команд

```

```

C:\asm\fasmw17121\Lab5Exmp x + v
09-Oct-25 03:30 PM      118 Lab4Exmpl3.com
09-Oct-25 02:43 PM      155 Lab4Exmpl3.prt
09-Oct-25 03:30 PM      1,686 Lab4Exmpl3_Main.asm
14-Oct-25 02:38 PM      74 Lab4Task1.com
14-Oct-25 08:06 AM      153 Lab4Task1.prt
14-Oct-25 02:38 PM      1,006 Lab4Task1_Main.asm
02-Nov-25 02:27 PM      108 Lab4Task2.com
02-Nov-25 12:21 PM      153 Lab4Task2.prt
02-Nov-25 02:27 PM      1,316 Lab4Task2_Main.asm
09-Oct-25 08:49 PM      83 LAB4TA~1.TR
02-Nov-25 02:25 PM      92 LAB4TA~4.TR
02-Nov-25 05:36 PM      1,536 Lab5Exmpl1.exe
02-Nov-25 04:51 PM      155 Lab5Exmpl1.prt
02-Nov-25 05:36 PM      441 Lab5Exmpl1_Main.asm
08-Mar-13 03:59 PM      1,820 LICENSE.TXT
04-Sep-25 06:00 PM      238 Main.asm
21-Feb-96 05:00 AM      120,853 rtm.exe
04-Sep-25 06:00 PM      36 Second_Project.com
05-Sep-25 03:10 PM      133 Second_Project.prt
04-Sep-25 10:35 AM      <DIR> SOURCE
21-Feb-96 05:00 AM      773,468 td.exe
02-Nov-25 02:08 PM      83 TD.TR
02-Nov-25 02:25 PM      691 TDCONFIG.TD
21-Feb-96 05:00 AM      158,483 tdhelp.tdh
21-Feb-96 05:00 AM      136,386 tdinst.exe
04-Sep-25 10:35 AM      <DIR> TOOLS
27-May-14 12:12 PM      15,431 WHATSNEW.TXT
86 File(s)      2,062,951 bytes
6 Dir(s)       47,607,451,648 bytes free
Press any key to continue . . .

```

Робота програми

### Завдання на 4 бали

1. Виконати завдання на 3 бали.
2. Виконати завдання згідно свого варіанту із таблиці 5.1 використовуючи тільки функції із мови С.

6, 16, 26

Відкрити будь-який txt файл  
та вивести інформацію на  
екран.

```

format PE console 4.0
include 'win32a.inc'
entry start
section '.code' code readable executable
start:
    cinvoke fopen, filename, type
    mov     ebx, eax
.nextLine:
    cinvoke fgets, buf, 255, ebx
    test    eax, eax
    jz      .done           ; якщо NULL -> кінець файлу
    cinvoke printf, fmt, buf
    jmp     .nextLine
.done:
    cinvoke fclose, ebx
    invoke  getch
    cinvoke exit, 0
section '.data' data readable writeable
type     db 'r',0
filename db 'file.txt',0
fmt      db '%s',13,10,0       ; формат для printf (додає
новий рядок)

section '.bss' readable writeable
buf rb 256                  ; буфер: 255 символів + '\0'

data import
library msvcrt,'MSVCRT.DLL', \
kernel32,'KERNEL32.DLL', \
user32,'USER32.DLL'

import msvcrt, \
puts,'puts', \
scanf,'scanf', \
printf,'printf', \
fopen,'fopen', \
fgets,'fgets', \
fclose,'fclose', \
fseek,'fseek', \
_getch,'_getch', \
exit,'exit'
end data

```

```

C:\asm\fasmw17121\Lab5Exmp: ~ + v
5 Лабораторна робота
Тема: «Використання WinAPI для створення програм на мові Assembler»
Мета роботи: навчитися використовувачи команди WinAPI для створення
найпростіших програм.

Хід роботи
Завдання на З бали
1. Відкомпілювати та відлагодити приклади представлени у роботі. Пояснити їх роботу, стани реєстрів та принципи роботи команд.

Приклад 1
format PE GUI 4.0
entry start
    include 'win32a.inc'
section '.data' data readable writable
    message db 'Hello World!',0
    title db 'Моя перша програма!',0
    section '.code' code readable executable
start:
push MB_OK
push title
push message
push 0
call [MessageBoxA]
close:
push 0
call [ExitProcess]

section '.idata' import data readable writeable
library kernel,'KERNEL32.DLL',\
user,'USER32.DLL'
import kernel, \
ExitProcess, ExitProcess'
import user, \
MessageBoxA, 'MessageBoxA'
Приклад 2
include 'include/win32ax.inc'
.code
main:
    invoke MessageBox,NULL,'Hello, World','Fasm message box:',MB_OK

```

## Робота програми

### Завдання на 5 балів

1. Виконати завдання на 3 та на 4 бали.
2. Виконати завдання згідно свого варіанту із таблиці 5.2 використовуючи тільки функції WinAPI.

6, 16, 26	Перемістити курсор миші на 300px ліворуч та на 100px вгору від поточного місцезнаходження. Змінити іконку курсору.
-----------	--------------------------------------------------------------------------------------------------------------------

```

format PE console 4.0
include 'win32a.inc'
entry start

```

```

IDC_HAND equ 32649 ; курсор "рука"
OCR_NORMAL equ 32512 ; стандартний курсор
SPI_SETCURSORS equ 0x57 ; команда для відновлення
стандартних курсорів

```

```
section '.code' code readable executable
start:
    ; отримати поточні координати курсора
    invoke GetCursorPos, point

    mov eax, [point.x]
    sub eax, 300
    mov [point.x], eax

    mov eax, [point.y]
    sub eax, 100
    mov [point.y], eax

    ; перемістити курсор у нову позицію
    invoke SetCursorPos, [point.x], [point.y]

    ; завантажити стандартний курсор "рука" і зробити його
    ; системним
    invoke LoadCursor, 0, IDC_HAND
    invoke SetSystemCursor, eax, OCR_NORMAL

    invoke getch                      ; чекати клавішу

    ; повернути стандартні курсори
    invoke SystemParametersInfo, SPI_SETCURSORS, 0, 0, 0

    invoke ExitProcess, 0

section '.data' data readable writeable
point POINT <>

data import
    library kernel32,'KERNEL32.DLL', \
            user32,'USER32.DLL', \
            msvcrt,'MSVCRT.DLL'

    import kernel32, \
            ExitProcess,'ExitProcess'

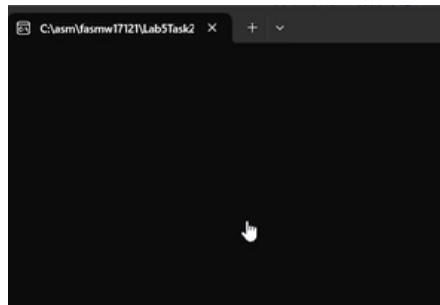
    import user32, \
            GetCursorPos,'GetCursorPos', \
```

```

SetCursorPos, 'SetCursorPos', \
LoadCursor, 'LoadCursorA', \
SetSystemCursor, 'SetSystemCursor', \
SystemParametersInfo, 'SystemParametersInfoA'

import msvcrt, \
    getch, '_getch'
end data

```



Робота програми

### Контрольні запитання

#### 1. Що таке WinAPI?

WinAPI – це набір функцій, констант і структур, що надаються операційною системою Windows для розробки програм, які взаємодіють із її компонентами (вікна, файли, процеси, пристрої тощо).

#### 2. Яка різниця між com та exe програмою?

СОМ-програма має фіксований розмір (до 64 КБ), завантажується в пам'ять як один сегмент і не містить заголовка. EXE-програма може мати більший розмір, складається з кількох сегментів (коду, даних, стеку) і має заголовок із службовою інформацією.

#### 3. Що таке функція, процедура та підпрограма?

Функція, процедура та підпрограма – це окремі частини програми, які виконують певне завдання. Процедура не повертає значення, функція – повертає результат через регистр або оператор return. Підпрограма – загальний термін, що охоплює і функції, і процедури.

#### 4. Що таке dll? Які особливості її використання?

DLL (Dynamic Link Library) – це динамічна бібліотека, яка містить функції чи ресурси, що можуть використовуватись кількома програмами одночасно. Її особливість – спільне завантаження в пам'ять, економія ресурсів і можливість оновлення окремих модулів без перекомпіляції основної програми.

#### 5. Основні принципи використання функцій мови С.

Функції мови С отримують параметри у визначеній послідовності, повертають результат через регистр EAX і зазвичай містяться в бібліотеках, наприклад msvcrt.dll. В асемблері їх виклик здійснюється командою invoke, при цьому параметри передаються через стек.