

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з комп'ютерного практикуму № 1 з дисципліни

«Системне програмне забезпечення»

«Створення програм на асемблері»

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив                      Лісовиченко Олег Іванович

Київ 2023

## Комп'ютерний практикум 1

### Створення програм на асемблері

#### Постановка завдання.

1. Для програми, наведеної нижче, створити файл типу .asm.
2. Скомпілювати програму, включивши потрібні опції для налагоджувача та створення файлу лістингу типу .lst.
3. Ознайомитись зі структурою файлу .lst.
4. Скомпонувати .obj-файл програми. Включити опції для налагодження та створення .map-файлу.
5. Занести до звіту адреси початку та кінця всіх сегментів з .map-файлу.
6. Завантажити до налагоджувача td.exe одержаний .exe-файл програми.
7. У вікні CPU у полі DUMP знайти початкову адресу сегмента даних та записати його до звіту. Знайти масиви SOURCE та DEST.
8. У покроковому режимі за допомогою клавіші F7 виконати програму. Одержані результати у масиві DEST показати викладачеві.

**Програма.**

```
STSEG SEGMENT PARA STACK "STACK"
```

```
DB 64 DUP ( "STACK" )
```

```
STSEG ENDS
```

```
DSEG SEGMENT PARA PUBLIC "DATA"
```

```
SOURCE DB 10, 20, 30, 40
```

```
DEST DB 4 DUP ( "?" )
```

```
DSEG ENDS
```

```
CSEG SEGMENT PARA PUBLIC "CODE"
```

```
MAIN PROC FAR
```

```
ASSUME CS: CSEG, DS: DSEG, SS: STSEG
```

```
; адреса повернення
```

```
PUSH DS
```

```
MOV AX, 0 ; або XOR AX, AX
```

```
PUSH AX
```

```
; ініціалізація DS
```

```
MOV AX, DSEG
```

```
MOV DS, AX
```

```
; обнуляємо масив
```

```
MOV DEST, 0
```

```
MOV DEST+1, 0
```

```
MOV DEST+2, 0
```

```
MOV DEST+3, 0
```

```
; пересилання
```

```
MOV AL, SOURCE
MOV DEST+3, AL
MOV AL, SOURCE+1
MOV DEST+2, AL
MOV AL, SOURCE+2
MOV DEST+1, AL
MOV AL, SOURCE+3
MOV DEST, AL
RET
MAIN ENDP
CSEG ENDS
END MAIN
```

**Введені та отримані результати.**

**Вміст .lst файлу**

Turbo Assembler Version 2.51      03/06/23 14:48:20      Page 1

lab1.ASM

```
1 0000                    STSEG SEGMENT PARA STACK "STACK"
2 0000 40*(53 54 41 43 4B) DB 64 DUP ( "STACK" )
3 0140                    STSEG ENDS
```

```

4 0000          DSEG SEGMENT PARA PUBLIC "DATA"

5 0000 0A 14 1E 28  SOURCE DB 10, 20, 30, 40

6 0004 04*(3F)          DEST DB 4 DUP ( "?" )

7 0008          DSEG ENDS

8 0000          CSEG SEGMENT PARA PUBLIC "CODE"

9 0000          MAIN PROC FAR

10          ASSUME CS: CSEG, DS: DSEG,      SS:
STSEG

11          ; адреса повернення

12 0000 1E          PUSH DS

13 0001 B8 0000          MOV AX, 0 ; або  XOR AX,  AX

14 0004 50          PUSH AX

15          ; ініціалізація DS

16 0005 B8 0000s      MOV AX, DSEG

17 0008 8E D8          MOV DS, AX

18          ; обнуляємо масив

19 000A C6 06 0004r 00  MOV DEST, 0

20 000F C6 06 0005r 00  MOV DEST+1, 0

21 0014 C6 06 0006r 00  MOV DEST+2, 0

22 0019 C6 06 0007r 00  MOV DEST+3, 0

23          ; пересилання

24 001E A0 0000r      MOV AL, SOURCE

25 0021 A2 0007r      MOV DEST+3, AL

26 0024 A0 0001r      MOV AL, SOURCE+1

```

27 0027 A2 0006r	MOV DEST+2, AL
28 002A A0 0002r	MOV AL, SOURCE+2
29 002D A2 0005r	MOV DEST+1, AL
30 0030 A0 0003r	MOV AL, SOURCE+3
31 0033 A2 0004r	MOV DEST, AL
32 0036 CB	RET
33 0037	MAIN ENDP
34 0037	CSEG ENDS
35	END MAIN

Turbo Assembler Version 2.51      03/06/23 14:48:20      Page 2

## Symbol Table

Symbol Name	Type	Value
??DATE	Text	"03/06/23"
??FILENAME	Text	"lab1 "
??TIME	Text	"14:48:20"
??VERSION	Number	0205
@CPU	Text	0101H
@CURSEG	Text	CSEG
@FILENAME	Text	LAB1

@WORDSIZE	Text 2
DEST	Byte DSEG:0004
MAIN	Far CSEG:0000
SOURCE	Byte DSEG:0000

Groups & Segments	Bit Size Align	Combine Class
CSEG	16 0037 Para	Public CODE
DSEG	16 0008 Para	Public DATA
STSEG	16 0140 Para	Stack STACK

### **Вміст .map файлу.**

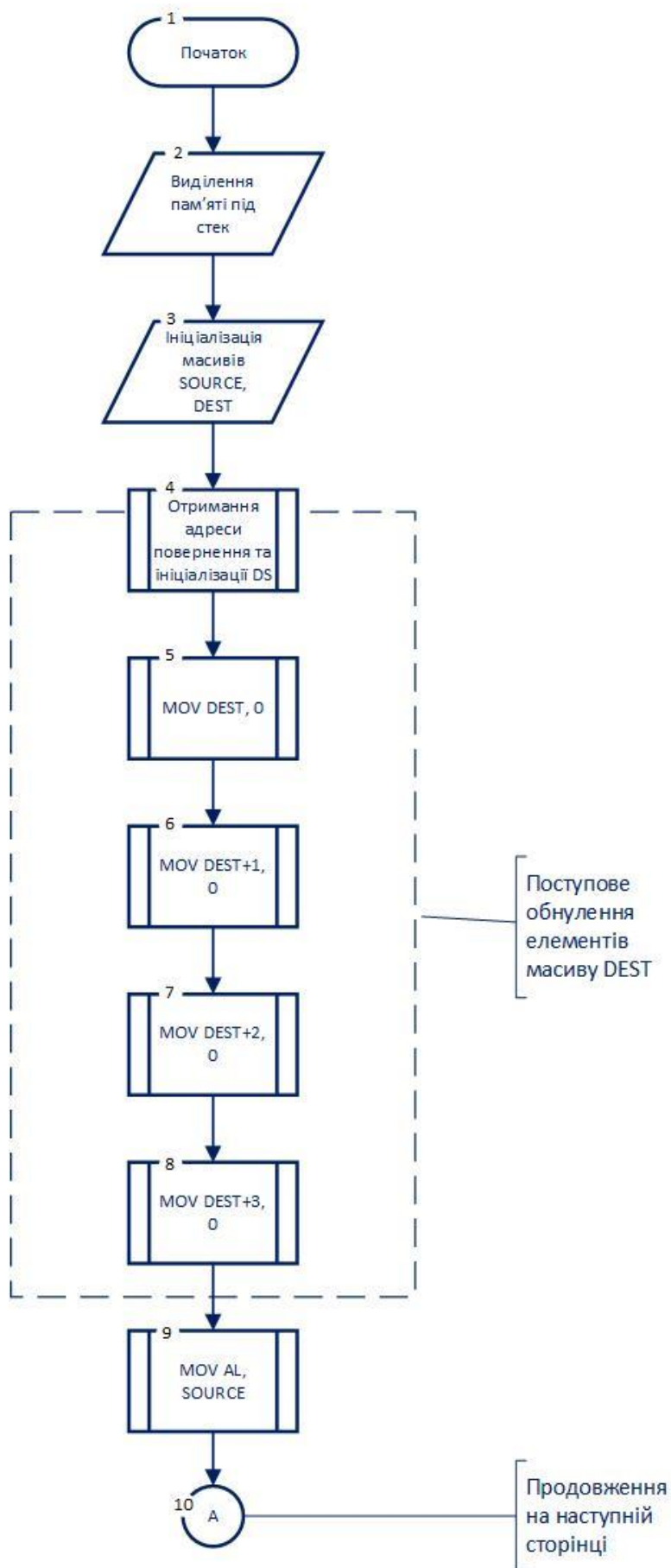
Start	Stop	Length	Name	Class
00000H	0013FH	00140H	STSEG	STACK
00140H	00147H	00008H	DSEG	DATA
00150H	00186H	00037H	CSEG	CODE

Address	Publics by Name
---------	-----------------

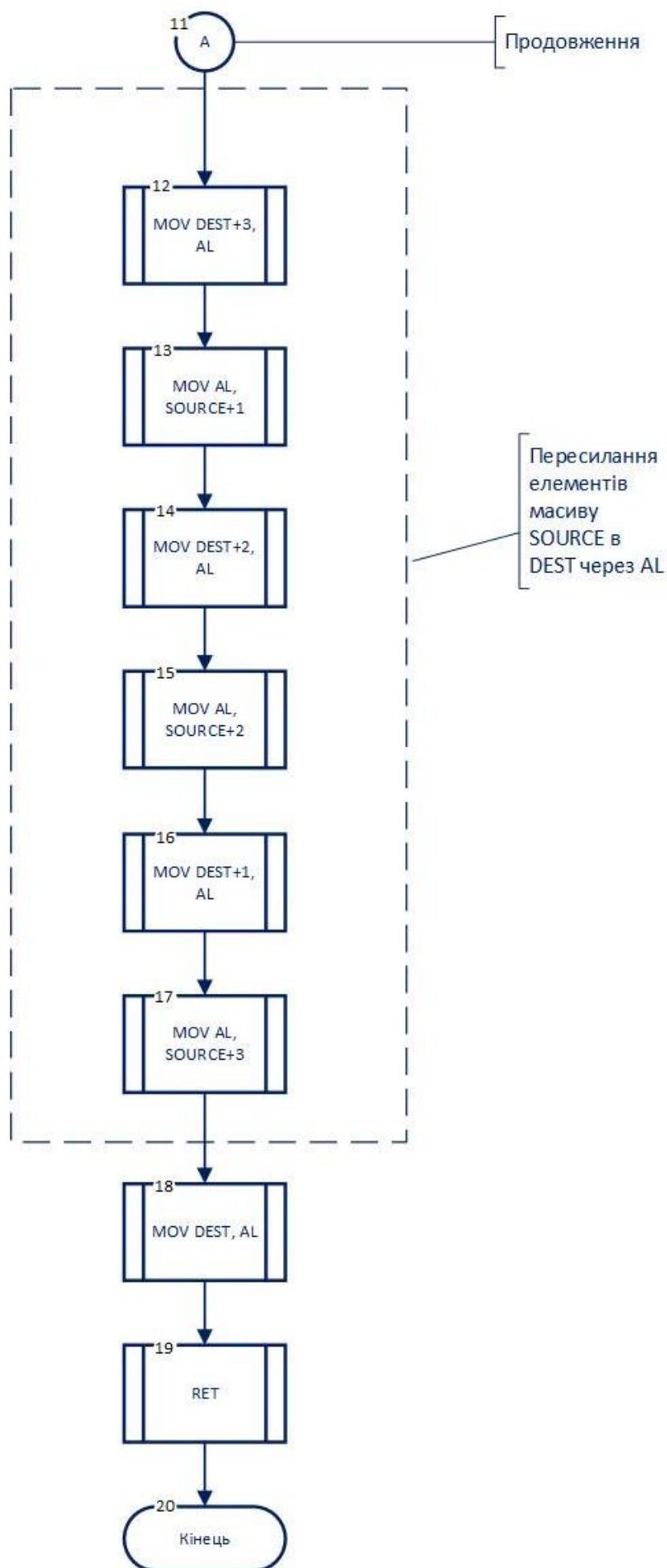
Address	Publics by Value
---------	------------------

Program entry point at 0015:0000

## СХЕМА ФУНКЦІОНУВАННЯ ПРОГРАМИ







## Виконання програми

## Вікно DUMP

```

ds:0000 0A 14 1E 28 3F 3F 3F 3F 0A (????
ds:0008 00 00 00 00 00 00 00 00
ds:0010 1E B8 00 00 50 B8 C1 44 P7 1D
ds:0018 8E D8 C6 06 04 00 00 C6 A7 1D

```

До виконання програми:

Масив SOURCE:

```

ds:0000 0A 14 1E 28 3F 3F 3F 3F 0A (????
ds:0008 00 00 00 00 00 00 00 00
ds:0010 1E B8 00 00 50 B8 C1 44 P7 1D
ds:0018 8E D8 C6 06 04 00 00 C6 A7 1D

```

Масив DEST:

```

ds:0000 0A 14 1E 28 3F 3F 3F 3F 0A (????
ds:0008 00 00 00 00 00 00 00 00
ds:0010 1E B8 00 00 50 B8 C1 44 P7 1D
ds:0018 8E D8 C6 06 04 00 00 C6 A7 1D

```

Після виконання програми:

Масив DEST:

```

ds:0000 0A 14 1E 28 28 1E 14 0A 0A ((A70
ds:0008 00 00 00 00 00 00 00 00
ds:0010 1E B8 00 00 50 B8 C1 44 P7 1D
ds:0018 8E D8 C6 06 04 00 00 C6 A7 1D

```

**Висновок.**

1. У текстовому редакторі було створено файл типу .asm з вихідним кодом програми відповідно до завдання;
2. Скомпілював програму, включивши потрібні опції для налагоджувача та створив файл лістингу типу .lst;
3. Ознайомився зі структурою .lst-файлу, розглянув структуру машинних команд;
4. Скомпонував .obj файл програми, включивши опції для налагодження та створення .map-файлу;
5. Відкрив файл карти пам'яті (.map-файл) та заніс до звіту адреси початку та кінця всіх сегментів програми;
6. Завантажив програму налагоджувача td.exe та одержаний .exe-файл програми;
7. У налагоджувачі відкрив вікно CPU, у полі DUMP знайшов початкові адреси сегментів даних, знайшов масиви SOURCE та DEST, подивився на значення елементів масивів;
8. У покроковому режимі за допомогою клавіші F7 виконав програму, програма коректно виконує задачу.