

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіти до комп'ютерних практикумів дисципліни

«Системне програмне забезпечення»

Прийняв
доцент кафедри ІІІ
Лісовиченко О.І.
“26” травня 2023 р.

Виконав
Студент групи ІІІ-11
Прищеп В.С.

Київ-2023

Комп'ютерний практикум №1

Тема: Створення програм на асемблері

Завдання:

1. Для програми, наведеної вище, створити файл типу .asm. Ця програма не має засобів виводу даних, тому правильність її виконання треба перевірити за допомогою td.exe.
2. Скомпілювати програму, включивши потрібні опції для налагоджувача та створення файлу лістингу типу .lst.
3. Ознайомитись зі структурою файлу .lst. За вказівкою викладача, для певної команди асемблера розглянути структуру машинної команди і навести її у звіті.
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. Дані у масиві SOURCE подаються у шістнадцятковій системі.
8. У покроковому режимі за допомогою клавіші F7 виконати програму. Одержані результати у масиві DEST показати викладачеві.

Текст програми:

```
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
;adress of return
push ds
mov ax, 0
push ax
;initialisation of ds
mov ax, dseg
mov ds, ax
; clearing of the array
mov dest, 0
mov dest+1, 0
mov dest+2, 0
mov dest+3, 0
```

```

; sending
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 4.0 03/26/23 14:46:43 Page 1
Lab1.asm

```

1 0000          stseg segment para    stack "stack"
2 0000 40*(73 74 61 63 6B) db    64 dup ("stack")
3 0140          stseg ends
4
5 0000          dseg segment para public "data"
6 0000 0A 14 1E 28 source db 10, 20, 30, 40
7 0004 04*(3F)          dest db 4 dup ("??")
8 0008          dseg ends
9
10 0000         cseg segment para public "code"
11 0000         main proc far
12             assume cs:      cseg, ds: dseg,      ss: stseg
13             ;adress of      return
14 0000 1E             push ds
15 0001 B8 0000             mov ax, 0
16 0004 50             push ax
17             ;initialisation of      ds
18 0005 B8 0000s             mov ax, dseg
19 0008 8E D8             mov ds, ax
20             ; clearing of the array
21 000A C6 06 0004r 00      mov dest, 0
22 000F C6 06 0005r 00      mov dest+1, 0
23 0014 C6 06 0006r 00      mov dest+2, 0

```

```

24 0019 C6 06 0007r 00      mov dest+3, 0
25                          ; sending
26 001E A0 0000r          mov al, source
27 0021 A2 0007r          mov dest+3, al
28 0024 A0 0001r          mov al, source+1
29 0027 A2 0006r          mov dest+2, al
30 002A A0 0002r          mov al, source+2
31 002D A2 0005r          mov dest+1, al
32 0030 A0 0003r          mov al, source+3
33 0033 A2 0004r          mov dest, al
34
35 0036 CB               ret
36 0037                 main endp
37 0037                 cseg ends
38                     end main

```

Turbo Assembler Version 4.0 03/26/23 14:46:43 Page 2
Symbol Table

Symbol Name	Type Value
??DATE	Text "03/26/23"
??FILENAME	Text "Lab1"
??TIME	Text "14:46:43"
??VERSION	Number 0400
@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:

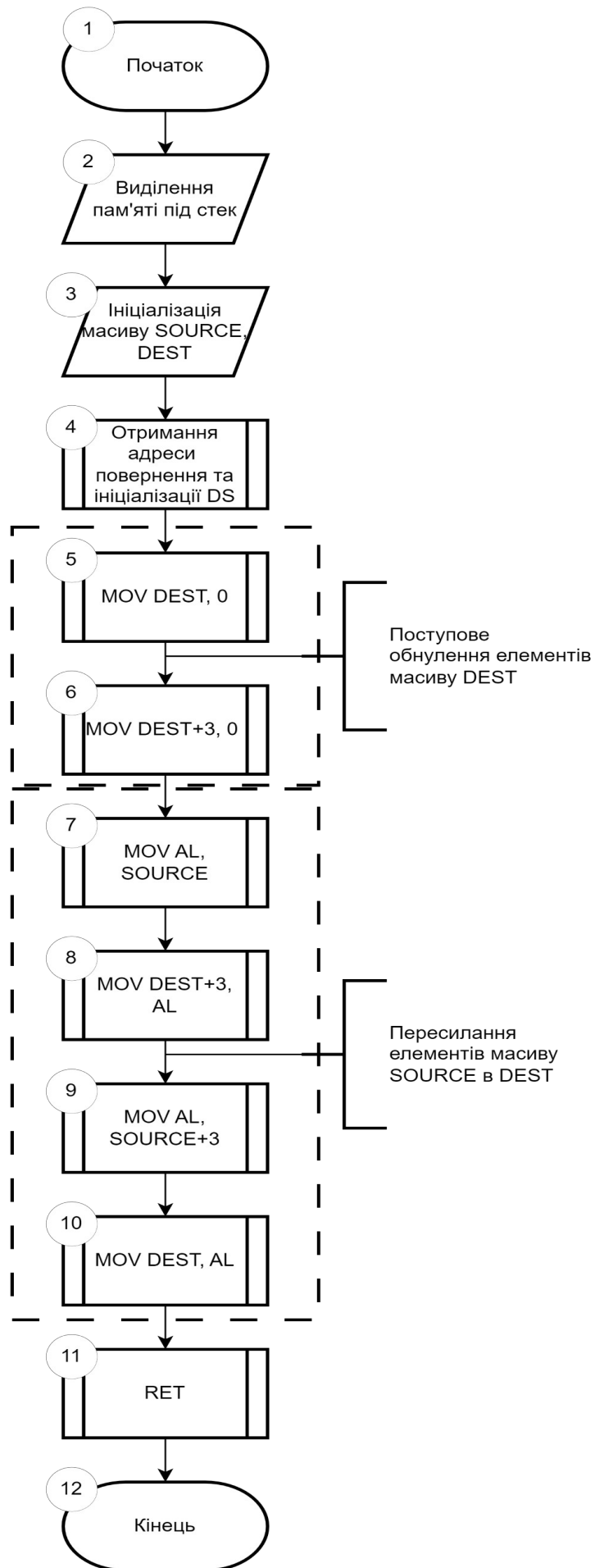
Start	Stop	Length	Name	Class
00000H	0013FH	00140H	STSEG	STACK
00140H	00147H	00008H	DSEG	DATA
00150H	00186H	00037H	CSEG	CODE

Address Publics by Name

Address Publics by Value

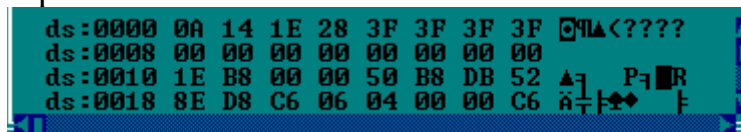
Program entry point at 0015:0000

Схема функціонування програми:



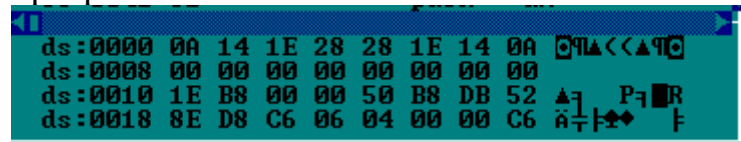
Вікно DUMP:

До виконання програми:



```
ds:0000 0A 14 1E 28 3F 3F 3F 3F 0A <????
ds:0008 00 00 00 00 00 00 00 00 00
ds:0010 1E B8 00 00 50 B8 DB 52  P R
ds:0018 8E D8 C6 06 04 00 00 C6  F F
```

Після виконання програми:



```
ds:0000 0A 14 1E 28 28 1E 14 0A 0A <<A
ds:0008 00 00 00 00 00 00 00 00 00
ds:0010 1E B8 00 00 50 B8 DB 52  P R
ds:0018 8E D8 C6 06 04 00 00 C6  F F
```

У рядку ds:0000 перші чотири байти належать масиву SOURCE, а останні чотири — DEST. До виконання програми у масиві DEST всі байти містять символи «?», а після виконання байти зеркально (у зворотній послідовності) дублюють вміст масиву SOURCE.

Висновок:

1. В текстовому редакторі було створено файл типу .asm.
2. Скомпілював програму, включивши потрібні опції для налагоджувача та створення файлу лістингу типу .lst.
3. Ознайомився зі структурою файлу .lst. Розглянув структури машинних команд.
4. Після усунення помилок, скомпонував .obj-файл програми, включивши опції для налагодження та створення .map-файлу.
5. Відкрив файл карти пам'яті (.map-файл) та подивився на адреси початку та кінця всіх сегментів програми.
6. Завантажив програму налагоджувача td.exe та мій одержаний .exe-файл програми.
7. У вікні CPU у полі DUMP подивився на початкову адресу сегмента даних. В сегменті даних знайшов масиви SOURCE та DEST. Дані у масиві SOURCE подаються у шістнадцятковій системі.
8. У покроковому режимі за допомогою клавіші F7 виконав програму. Програма коректно виконує поставлену задачу.