

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.

“20” лютого 2024р.

Виконав

Студент групи ІІІ-24

Піддубний Б.С.

Київ 2024

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:

; Опис сегменту стеку

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

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

```
STSEG ENDS
```

; Опис сегменту даних

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

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

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

; Опис сегменту коду

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

; Код основної функції

```
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
```

; Обнулення масиву

```
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 4.1 20/02/24 01:55:32 Page 1

test.asm

```
1                                ; Опис  сегменту стеку
2 0000                          STSEG SEGMENT PARA  STACK "STACK"
```

```

3 0000 40*(53 54 41 43 4B)          DB 64 DUP ( "STACK" )
4 0140                               STSEG ENDS
5
6                                   ; Опис сегменту даних
7 0000                               DSEG SEGMENT PARA PUBLIC "DATA"
8 0000 0A 14 1E 28                   SOURCE DB 10,20, 30, 40
9 0004 04*(3F)                       DEST   DB 4 DUP ( "?" )
10 0008                              DSEG ENDS
11
12                                   ; Опис сегменту коду
13 0000                               CSEG SEGMENT PARA PUBLIC "CODE"
14
15                                   ; Код основної функції
16 0000                               MAIN PROC FAR
17                                   ASSUME CS: CSEG, DS: DSEG,
SS: STSEG
18
19                                   ; Адреса повернення
20 0000 1E                           PUSH DS
21 0001 B8 0000                       MOV AX, 0
; або XOR AX, AX
22 0004 50                           PUSH AX
23
24                                   ; Ініціалізація DS
25 0005 B8 0000s                       MOV AX, DSEG
26 0008 8E D8                         MOV DS, AX

```

```

27
28                                ; Обнулення масиву
29 000A C6 06 0004r 00          MOV  DEST, 0
30 000F C6 06 0005r 00          MOV  DEST+1, 0
31 0014 C6 06 0006r 00          MOV  DEST+2, 0
32 0019 C6 06 0007r 00          MOV  DEST+3, 0
33
34                                ; Пересилання
35 001E A0 0000r                MOV  AL, SOURCE
36 0021 A2 0007r                MOV  DEST+3, AL
37 0024 A0 0001r                MOV  AL, SOURCE+1
38 0027 A2 0006r                MOV  DEST+2, AL
39 002A A0 0002r                MOV  AL, SOURCE+2
40 002D A2 0005r                MOV  DEST+1, AL
41 0030 A0 0003r                MOV  AL, SOURCE+3
42 0033 A2 0004r                MOV  DEST, AL
43
44 0036 CB                      RET
45 0037                        MAIN ENDP
46 0037                        CSEG ENDS
47                        END MAIN

```

Symbol Table

Symbol Name	Type Value
??DATE	Text "20/02/24"
??FILENAME	Text "test "
??TIME	Text "01:55:32"
??VERSION	Number 040A
@CPU	Text 0101H
@CURSEG	Text CSEG
@FILENAME	Text TEST
@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

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

Вікно DUMP:

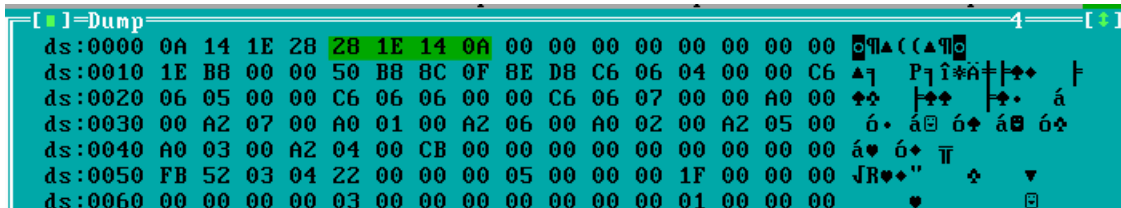
Массив SOURCE:

```
[ ]=Dump 4 [ ]
```

ds:0000	0A 14 1E 28	3F 3F 3F 3F	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
ds:0010	1E B8 00 00	50 B8 8C 0F	8E D8 C6 06	04 00 00 C6	A7 Pj i?Ä?	†♦ †	†
ds:0020	06 05 00 00	C6 06 06 00	00 00 C6 06	07 00 00 A0	♣♣ †♣ †♣	á	
ds:0030	00 A2 07 00	A0 01 00 A2	06 00 A0 02	00 00 A2 05	ó. á▯ ó♣	á▯ ó♣	
ds:0040	A0 03 00 A2	04 00 CB 00	00 00 00 00	00 00 00 00	á ♡ ó ♢	π	
ds:0050	FB 52 03 04	22 00 00 00	05 00 00 00	1F 00 00 00	JR♦ " ♣	▼	
ds:0060	00 00 00 00	03 00 00 00	00 00 00 00	00 00 01 00	♥	▯	

[illegible]

Масив DEST:



Висновок:

1. Я створив файл типу .asm.
2. Скомпілював програму, врахував потрібні опції для налагоджувача та створення файлу типу .lst.
3. Ознайомився зі структурою файлу .lst. Розглянув структури машинних команд.
4. Після усунення помилок, скомпонував .obj файл програми, врахував 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 налагодження td.exe та .exe файл програми.
7. У вікні CPU в 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.