**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.

1. Занести до звіту адреси початку та кінця всіх сегментів з .map-файлу.
2. Завантажити до налагоджувача td.exe одержаний .exe-файл програми.
3. 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.
4. У покроковому режимі за допомогою клавіші F7 виконати програму. Одержані результати у масиві DEST показати викладачеві.

Текст програми

; описание сегмента стека

STSEG SEGMENT PARA STACK "STACK" DB 64 DUP ("STACK")

STSEG ENDS

; описание сегмента данных

DSEG SEGMENT PARA PUBLIC "DATA" SOURCE DB 10, 20, 30, 40

DEST DB 4 DUP ( "?" ) DSEG ENDS

; описание сегмента кода

CSEG SEGMENT PARA PUBLIC "CODE"

; код основной функции

MAIN PROC FAR

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

; адрес возврата

PUSH DS

MOV AX, 0 ; або XOR AX, AX PUSH AX

; инициализация DS MOV AX, DSEG MOV DS, AX

; обнуление массива

MOV DEST, 0

MOV DEST+1, 0

MOV DEST+2, 0

MOV DEST+3, 0

; пересылка

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 09/10/12 21:47:30 Page 1 lab1.asm

|  |  |  |
| --- | --- | --- |
| 1 |  | ; описание сегмента стека |
| 2 0000 |  | STSEG SEGMENT PARA STACK "STACK" |
| 3 0000 | 40\*(53 54 | 41 43 4B) DB 64 DUP ("STACK") |
| 4 0140 |  | STSEG ENDS |
| 5 |  | ; описание сегмента данных |
| 6 0000 |  | DSEG SEGMENT PARA PUBLIC "DATA" |
| 7 0000 | 0A 14 1E 28 | SOURCE DB 10, 20, 30, 40 |
| 8 0004 | 04\*(3F) | DEST DB 4 DUP ( "?" ) |
| 9 0008 |  | DSEG ENDS |
| 10 |  | ; описание сегмента кода |
| 11 0000 |  | CSEG SEGMENT PARA PUBLIC "CODE" |
| 12 |  | ; код основной функции |
| 13 0000 |  | MAIN PROC FAR |
| 14 |  | ASSUME CS: CSEG, DS: DSEG, SS: STSEG |
| 15 |  | ; адрес возврата |
| 16 0000 | 1E | PUSH DS |
| 17 0001 | B8 0000 | MOV AX, 0 ; або XOR AX, AX |
| 18 0004 | 50 | PUSH AX |
| 19 |  | ; инициализация DS |
| 20 0005 | B8 0000s | MOV AX, DSEG |
| 21 0008 | 8E D8 | MOV DS, AX |
| 22 |  | ; обнуление массива |
| 23 000A | C6 06 0004r 00 | MOV DEST, 0 |
| 24 000F | C6 06 0005r 00 | MOV DEST+1, 0 |
| 25 0014 | C6 06 0006r 00 | MOV DEST+2, 0 |
| 26 0019 | C6 06 0007r 00 | MOV DEST+3, 0 |
| 27 |  | ; пересылка |
| 28 001E | A0 0000r | MOV AL, SOURCE |
| 29 0021 | A2 0007r | MOV DEST+3, AL |
| 30 0024 | A0 0001r | MOV AL, SOURCE+1 |
| 31 0027 | A2 0006r | MOV DEST+2, AL |

|  |  |  |  |
| --- | --- | --- | --- |
| 32 002A A0 0002r |  | MOV | AL, SOURCE+2 |
| 33 002D A2 0005r |  | MOV | DEST+1, AL |
| 34 0030 A0 0003r |  | MOV | AL, SOURCE+3 |
| 35 0033 A2 0004r  36  37 0036 CB | RET | MOV | DEST, AL |

38

39 0037 MAIN ENDP 40

41 0037 CSEG ENDS 42

43 END MAIN

\_Turbo Assembler Version 4.1 09/10/12 21:47:30 Page 2 Symbol Table

Symbol Name Type Value

??DATE Text "09/10/12"

??FILENAME Text "lab1 "

??TIME Text "21:47:30"

??VERSION Number 040A

@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

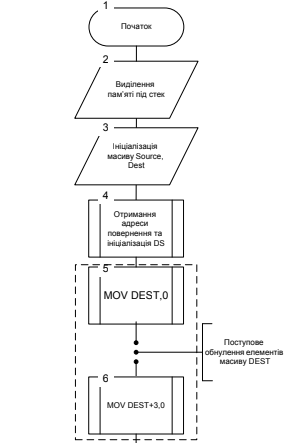
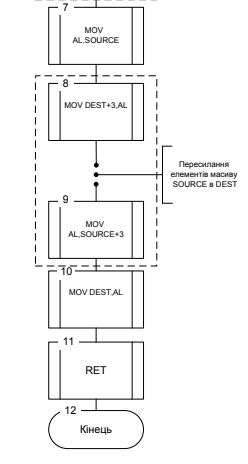
00000H 0013FH 00140H STSEG STACK

00140H 00147H 00008H DSEG DATA

00150H 00186H 00037H CSEG CODE

Program entry point at 0015:0000

**Схема функціонування програми**



**Вікно DUMP**



**До виконання програми:**

**Масив SOURCE:**

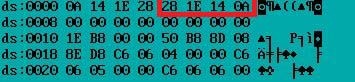


**Масив DEST**



**Після виконання програми**

**Масив DEST**



**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 памяті (.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 даниз 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.