**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Звіт до комп’ютерного практикума робіт з дисципліни

Системне програмування

*Виконав:*

студент групи ЗПІ-ЗП 61

Макіян Смбат Артурович

*Перевірив:*

Викладач

доцент, к.т.н.

Лісовиченко Олег Іванович

Київ – 2018

**Звіт з практикума №1**

**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 файла лістингу типу .lst.
3. Ознайомитись зі структурою файла .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

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

**push** DS

**xor** AX, AX

**push** AX

**mov** AX, DSEG

**mov** DS, AX

**mov** DEST,0

**mov** DEST+1, 0

**mov** DEST+2, 0

**mov** DEST+3, 0

**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

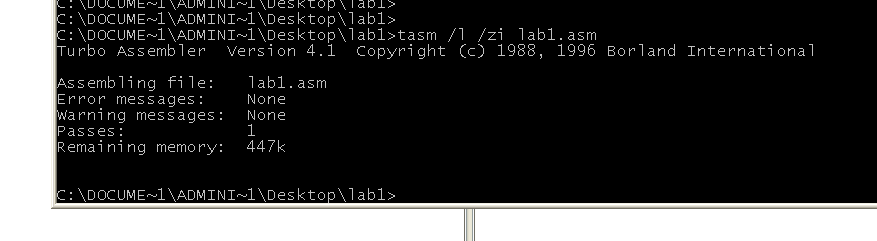
END MAIN

**Виконання:**

ПЗ, що використовуються для виконання даної лабораторної роботи адаптоване для використання в середовищі MS DOS, тому, було встановлено ОС Windows XP Mode 32 bit під VirtualBox.

Для створення .asm файлу використовуються редактор Sublime Text 3

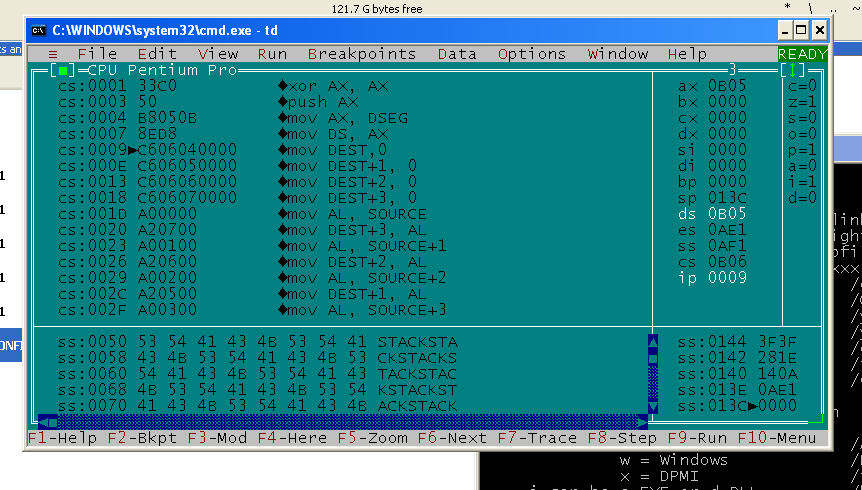
Для перевірки з використанням td.exe необхідно отримати скомпільовану виконувану програму в «.EXE», для цього необхідно створити файл типу «.OBJ» за допомогою TASM.EXE а потім з файлу «.OBJ» отримати «.EXE», за допомогою TLINK.EXE.



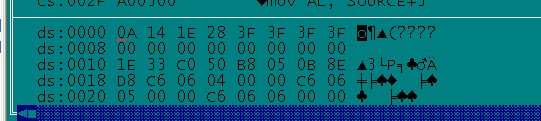
З контекстного меню відкриваємо виконуваний файл «.EXE», в даному випадку LAB1.EXE

Покроково виконавши дебаг програми за допомогою клавіши F7, можна зазначити зміни, що виникають у сегменті стеку SS;, куди записуються початкові значення регістрів DS та AX та зміни у сегменті даних DS: де вказано зміщення (у стеку даних) та значення у шістнадцятковому форматі (DS:0000 = 0A, DS:0001 = 14, DS:0002 = 1E, DS:0003 = 28). Що у свою чергу відповідає 10, 20, 30, 40 у десятковому форматі. Т.к. данні зі стеку можно отримати лиже з кінця, но останнє число (40) буде першим, тобто масив чисел будем у зворотному порядку.

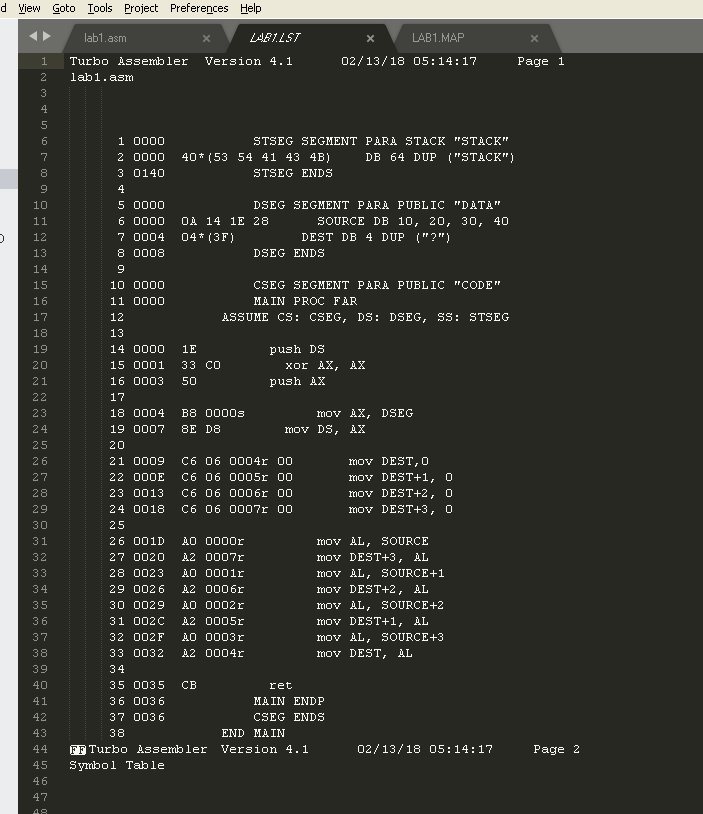
Для створення лістингу типу .lst. програми, необхідно додати параметр /l при створенні файлу «.OBJ».

З контекстного меню відкриваємо виконуваний файл «.EXE», в даному випадку LAB1.EXE

Покроково виконавши дебаг програми за допомогою клавіши F7, можна зазначити зміни, що виникають у сегменті стеку SS;, куди записуються початкові значення регістрів DS та AX та зміни у сегменті даних DS: де вказано зміщення (у стеку даних) та значення у шістнадцятковому форматі (DS:0000 = 0A, DS:0001 = 14, DS:0002 = 1E, DS:0003 = 28). Що у свою чергу відповідає 10, 20, 30, 40 у десятковому форматі. Т.к. данні зі стеку можно отримати лиже з кінця, но останнє число (40) буде першим, тобто масив чисел будем у зворотному порядку.



Переглянемо вміст файлу LAB1.LST:

З малюнку видно, що «.LST» файл містить інформацію про версію Турбо Ассемблера, дату, час створення, позначку сторінки, назву вихідного файлу.

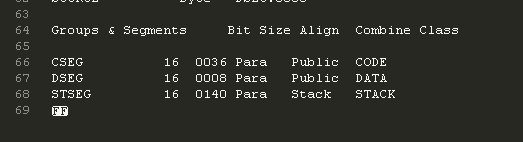
Далі міститься таблиця стовпців «вкладення», «номер\_строки», «зміщення», «машинний код», «вихідний\_код».

«вкладення» в данному випадку пусте, бо немає вкладених файлів, тощо.

«зміщення» вказує на зміщення поточного сегменту відповідно від початкової строки генерованої Турбо Ассемблером.

«машинний код» вказує фактичну послідовність шістнадцяткових значень розміром в байт або слово, які отримані із відповідної строки «вихідний\_код».

На другій сторінці містяться данні про групи та сегменти з інформацією про кількість біт, розмірність, вирівнювання, атрибут та клас Мінімальна кількість сегментів - 3.



«.MAP» містить інформацію про адреси початку та кінця всіх сегментів а також про розмір сегментів, що співпадає з даними із «.LST» файлу:

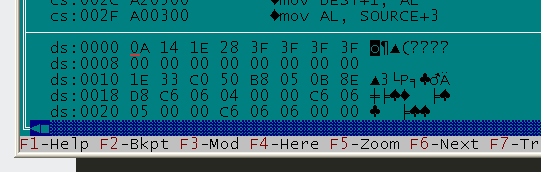


Початкова адреса сегмента даних DS:0000:

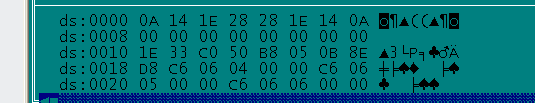
Вміст масивів SOURCE та DEST можна переглянути за допомогою сегмента даних:

Виходячи з малюнка, можна зробити висновок, що вхідний масив SOURCE був перенесений до масиву DEST у оберненому вигляді.

Було



Стало



**Висновок:**

Програма вирішує поставлене завдання. Теоретичні розрахунки відповідають отриманим. Програма працює правильно.