МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Лабораторна робота №4**

з дисципліни « Програмування на асемблері »

на тему: «УВЕДЕННЯ В МОВУ АСЕМБЛЕР  
(EMU8086, DEBUG, TASM, MASM)»

Виконав: студент 2 курсу групи № 622п

121 «Інженерія програмного забезпечення»

(код спеціальності)

Зайченко Я. І.

(ПІБ студента)

Прийняв: ст. викладач каф. 603

\_ Дем’яненко В. А.\_\_\_\_\_\_

(ПІБ викладача)

Національна шкала:

Кількість балів:

Харків – 2024

**Мета роботи:** ознайомитися з роботою різних Асемблерів при написанні програми мовою Асемблер.

Структура звіту

1. таблиця значень регістрів розробленої програми для Emu8086;
2. екранні форми виконання створеної програми для Emu8086;
3. екранні форми виконання створеної програми для Debug (звичайне й послідовне виконання);
4. екранні форми виконання дизасемблювання в Debug;
5. екранна форма створеної програми на машинному коді в шістнадцятковому редакторі;
6. висновок.

Виконання роботи

Створюємо новий COM файл з кодом поданим нижче:

MOV AH, 02h

MOV DL, 41h

INT 21h

INT 20h

В режимі відладки можемо відслідкувати зміни в реєстрах. Їх опис поданий в таблиці 1.

Таблиця 1 – таблиця значень регістрів розробленої програми для Emu8086;

|  |  |  |
| --- | --- | --- |
| **Крок** | **Зміна регістру** | **Пояснення** |
|  | AX(H) = 02 | AH (вищий байт регістру AX) отримує значення 02h. Це код служби INT 21h для виводу символу. |
| IP = 0102 | Інструкція INT 21h виконується, тому IP (вказівник інструкційного регістру) збільшується на 2. |
|  | DX(L) = 41 | DL (нижній байт регістру DX) отримує значення 41h, яке відповідає символу 'A'. |
| IP = 0104 | Інструкція INT 20h виконується, IP збільшується на 2. |
|  | CS = F400 | Сегментний регістр CS змінюється на F400h, що є адресою INT 20h у сегменті виходу (зазвичай DOS). |
| IP = 0200 | IP змінюється на адресу INT 20h в новому сегменті. |
| SS = FFF8 | Стековий сегментний регістр SS змінюється на FFF8h, щоб відведені дані могли бути збережені у стеку. |
|  | AX(L) = 41 | AL (нижній байт регістру AX) отримує значення 41h, яке відповідає символу 'A'. |
| IP = 0204 | IP збільшується на 2. |
|  | CS = 0700 | Повертаємось до вихідного сегменту CS. |
| IP = 0106 | IP змінюється на наступну інструкцію після INT 21h. |
| SP = FFFE | Вказівник стеку SP зменшується, що може означати видалення даних зі стеку. |
|  | CS = F400 | Сегментний регістр CS знову змінюється на F400h, повертаємось до INT 20h. |
| IP = 0150 | IP змінюється на адресу INT 20h у вихідному сегменті. |
| SP = FFF8 | Вказівник стеку SP змінюється, можливо, для відведення місця для збереження даних. |
|  | IP = 0154 | IP змінюється на наступну інструкцію після INT 20h. |

Результат виконання коду в емуляторі подано на рисунку 1

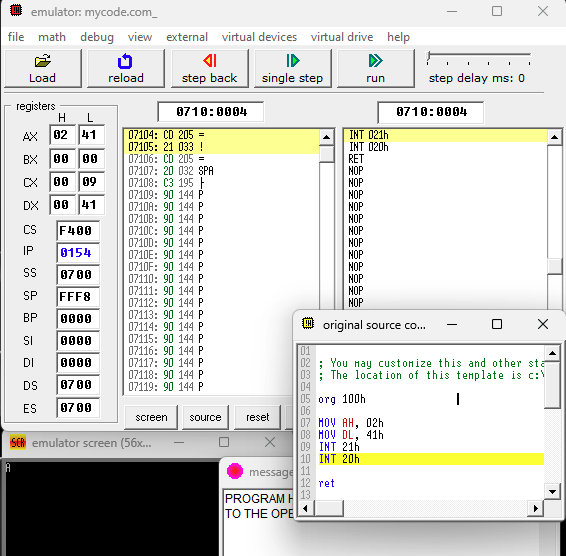


Рисунок 1 – екранні форми виконання створеної програми для Emu8086;

В програмі Debug вводимо спочатку флаг «-а». Та вводимо код програми. після пишемо команду «-g». Див. рисунок 2-3.

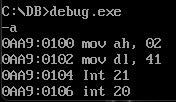


Рисунок 2 – екранна форма написання програми для Debug

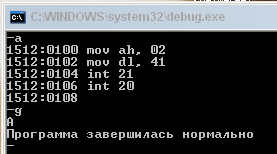


Рисунок 3 – екранна форма виконання створеної програми для Debug (звичайне виконання);

Для запуску коду в покроковому режимі повторюємо попередні кроки та вводимо команду «t».

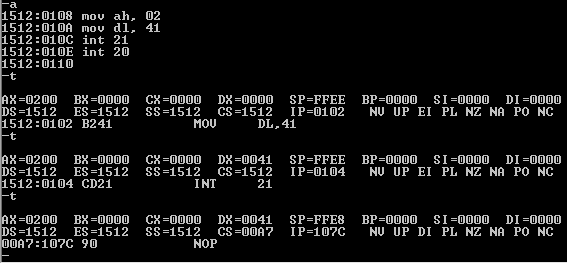


Рисунок 4 –кроки програми в Debug

Для завершення виконання коду необхідно ввести команду «g»

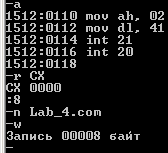


Рисунок 5 – створення \*.com файлу з коду програми

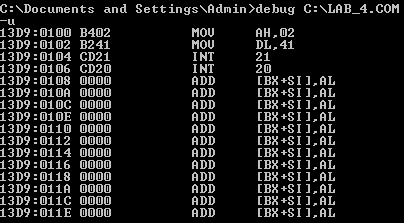


Рисунок 6 – результат дизасемблювання файлу \*.com з кодом програми

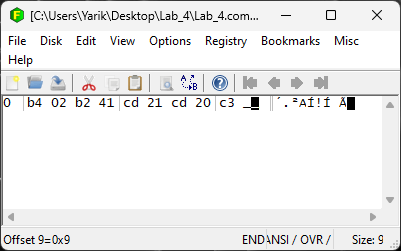


Рисунок 7 – файл \*.com в шістнадцятковому редакторі

Висновок

Під час виконання лабораторної роботи ознайомилися з роботою різних асемблерів, зокрема Emu8086 та Debug. Було написано програму мовою асемблер, яка виводила символ 'A' на екран, використовуючи системний виклик DOS INT 21h.

У роботі була розроблена таблиця значень регістрів для програми, виконаної в середовищі Emu8086. Також були створені екранні форми виконання програми для Emu8086 та Debug. У Debug було вивчено, як здійснюється послідовне виконання програми, а також як проводиться дизасемблювання.

Додатково був перевірений машинний код нашої програми в шістнадцятковому редакторі.

Ця лабораторна робота дозволила отримати практичні навички роботи з асемблером, а також вивчити процес виконання програм на низькому рівні, що було важливим для розуміння роботи комп'ютерних систем.