

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота №2

з дисципліни «Архітектура комп'ютера»

Варіант-3

Виконали: Студенти групи ІТ-01

Гончаренко А.А.

Чорній В.І.

Захищено з балом _____

Київ-2020

Мета: набутті впевнених знань і навичок з розробки ПЗ на Асемблері для управління відеопам'яттю з урахуванням знань архітектури IA-32 у real address mode.

Хід роботи

1. Виконати повний цикл розробки, тестування і налагодження програмного забезпечення.

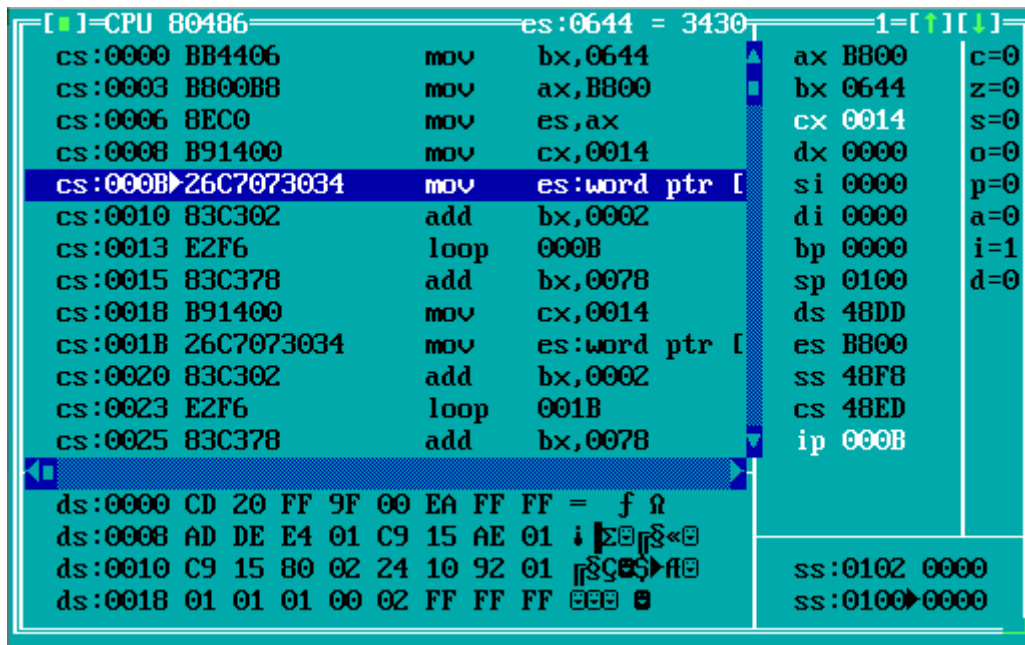
Для вирішення завдання був написаний код на Асемблері. (посилання на код [Чорнія](#) та [Гончаренка](#)).

На початку сегменту коду у регістрі bx задається початок прямокутника, у регістр es задається початок сегменту відеопам'яті через регістр ax та задаємо кількість повторень циклу, що виводить перший рядок.

```
CODESEG
Start:

    mov bx, 1604; задаємо початок прямокутника (80*2*10+2*2)
    mov ax, 0B800h; задаємо початок сегменту відеопам'яті
    mov es, ax
    mov cx, 20; задаємо кількість символів у 1 рядку
```

Результат виконання у турбодебагері:

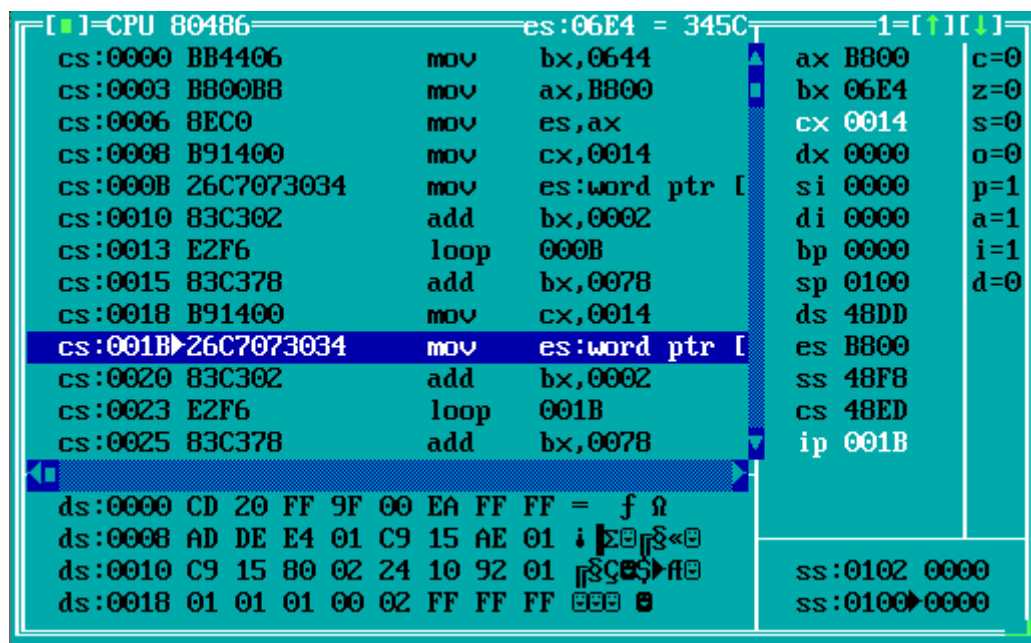


Наступним кроком виконується цикл виводу символів у рядок. У регістр es записується символ. Після цього зміщується адреса у відеопам'яті для запису

наступного символу. Після виконання циклу адреса у відеопам'яті зміщується на 120 для переносу на наступний рядок та задається кількість повторень наступного циклу.

```
my_ptr::створення циклу для 1 рядка
    mov [word es:bx], 3430h; запис симола у відео пам'ять
    add bx,2    ; зміщення адреси у відеопа'мяті
    Loop my_ptr ; виклик циклу
    add bx, 120; переведення рядка(80*2-20*2)
    mov cx,20; задаємо кількість символів у 1 рядку
```

Результат виконання у турбодебагері:



Аналогічно написано наступні 9 циклів для виводу усіх рядків.

Завершується програма записом у регістр ah функції виходу з програми, повернення вихідного коду та викликом функції завершення програми.

```
mov ah,04Ch; функція виходу з програми
mov al,0; повернення вихідного коду
int 21h; виклик функції 04Ch

end Start
```

Асемблюємо код командою tasm:

```
F:\>tasm laba2.asm
Turbo Assembler Version 3.2i Copyright (c) 1988, 1992 Borland International
Serial No:   Tester:

Assembling file:   laba2.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  470k
```

Лінкуємо .obj файл командою tlink:

```
F:\>tlink laba2.obj
Turbo Link Version 5.1 Copyright (c) 1992 Borland International
```

Запускаємо .exe файл:

```
F:\>tasm laba2.asm
Turbo Assembler Version 3.2i
Serial No:   Tester:

Assembling file:   laba2.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  470k

F:\>tlink laba2.obj
Turbo Link Version 5.1 Copyright (c) 1992 Borland International

F:\>laba2.exe
```

Висновки.

В ході лабораторної було набуто твердих навичок і знань технологічної основи розробки ПЗ на Асемблері для управління відеопам'яттю з урахуванням знань архітектури IA-32 у real address mode.