Міністерство освіти і науки України Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського" Факультет інформатики та обчислювальної техніки Кафедра автоматики та управління в технічних системах

Лабораторна робота 4 Архітектура комп'ютера

"Управління ходом виконання програми на асемблері apximeктури IA-32 (x86) у Real Adress Mode"

Виконали: Перевірив:

Студенти групи IT-01

Тимошенко Олексій Тонкий Михайло Бердник Ю.М. **Тема:** Управління ходом виконання програми на асемблері архітектури IA-32 (x86) у Real Adress Mode.

Мета: отримати основні навички управління ходом виконання програми.

Github: https://github.com/OlexiiT/asembler/tree/main/L5 https://github.com/Mikuyoki/Asembler/tree/main/L5

Хід роботи:

Згідно з нашим варіантом необхідно було заповнити у масиві ділянку 8х8 нашими датами народження починаючи з координат (7, 7). Для цього спочатку було оголошено масив array2Db:

```
array2Db
         db 7, 8, 7, 8, 7, 8, 7, 8, 7, 8, 7, 8, 7, 8
         db 7, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3
         db 7, 3, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3,
         db 7, 3, 3, 7, 3, 3, 3, 3, 3, 3, 3, 3,
         db 7, 3, 3, 3, 7, 3, 3, 3, 3, 3, 3, 3, 3,
         db 7, 3, 3, 3, 3, 7, 3, 3, 3, 3, 3, 3, 3, 3, 3,
         db 7, 3, 3, 3, 3, 7, 3, 3, 3, 3, 3, 3,
         db 7, 3, 3, 3, 3, 3, 7, 3, 3, 3, 3,
         db 7, 3, 3, 3, 3, 3, 3, 7, 3, 3, 3,
         db 7, 3, 3, 3, 3, 3, 3, 3, 7, 3, 3, 3,
         db 7, 3, 3, 3, 3, 3, 3, 3, 3, 7, 3, 3,
         db 7, 3, 3, 3, 3, 3, 3, 3, 3, 3, 7, 3, 3, 3,
         db 7, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 7,
         db 7, 8, 7, 8, 7, 8, 7, 8, 7, 8, 7, 8, 7, 8
```

Заповнення ділянки ми виконали за допомогою базово-індексної адресації:

```
-----Заповнення масиву датами народження-----
mov bx, 077h
                                                          ;Підготовка регістрів до базової адресації
mov cx, 4
wl:
                                                          ;Заповнення масиву першою датою
    xor si, si
                                                          ;Встановлюємо регістр рівним нулю
    mov al, [birthdays_arrayl+si]
                                                          ;Записуємо першу цифру дати
    mov [array2Db+bx+si], al
                                                          ;Записуємо цифру у масив
                                                          ;Збільшуємо лічильник на 1
    mov al, [birthdays_arrayl+si]
                                                          ;Повторюємо до кінця рядочка..
    mov [array2Db+bx+si], al
    inc si
    mov al, [birthdays_arrayl+si]
    mov [array2Db+bx+si], al
    inc si
    mov al, [birthdays_arrayl+si]
    mov [array2Db+bx+si], al
    inc si
    mov al, [birthdays arrayl+si]
    mov [array2Db+bx+si], al
    mov al, [birthdays arrayl+si]
    mov [array2Db+bx+si], al
    mov al, [birthdays_arrayl+si]
    mov [array2Db+bx+si], al
    inc si
    mov al, [birthdays_arrayl+si]
    mov [array2Db+bx+si], al
    add bx, 20h
    loop wl
mov bx, 087h
                                                      ;Підготовка регістрів до базової адресації
mov si, Oh
mov cx, 4
w2:
                                                     ;Заповнення масиву другою датою
   xor si, si
                                                      ;Встановлюємо регістр рівним нулю
   mov al, [birthdays_array2+si]
                                                      ;Записуємо першу цифру дати
   mov [array2Db+bx+si], al
                                                      :Записуемо цифру у масив
    inc si
                                                      ;Збільшуємо лічильник на 1
   mov al, [birthdays_array2+si]
                                                      ;Повторюємо до кінця рядочка..
   mov [array2Db+bx+si], al
    inc si
    mov al, [birthdays_array2+si]
    mov [array2Db+bx+si], al
    inc si
   mov al, [birthdays_array2+si]
    mov [array2Db+bx+si], al
    inc s
   mov al, [birthdays_array2+si]
    mov [array2Db+bx+si], al
    mov al, [birthdays_array2+si]
    mov [array2Db+bx+si], al
    mov al, [birthdays_array2+si]
    mov [array2Db+bx+si], al
    inc si
    mov al, [birthdays_array2+si]
    mov [array2Db+bx+si], al
    loop w2
```

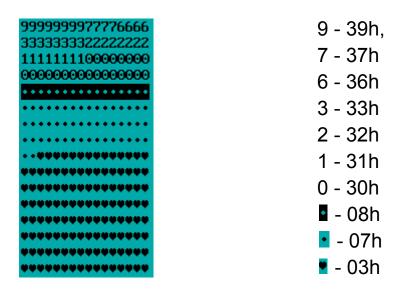
I отримали такий результат:

Також ми реалізували алгоритм сортування (за спаданням) даного масиву та застосували його:

Алгоритм:

```
------Сортування масиву-----
mov ex, 255
PROC sort
   loop1:
      mov bx, cx
       mov si, 0
   comp:
       mov al, [array2Db+si]
       mov dl, [array2Db+si+1]
       cmp al, dl
       jnb noswap
       mov [array2Db+si], dl
       mov [array2Db+si+1], al
   noswap:
       inc si
       dec bx
       jnz comp
       loop loopl
ENDP sort
```

Результат його роботи:



Висновки: під час роботи ми здобули навички управління ходом виконання програми.