

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

*Звіти до комп'ютерних практикумів з кредитного модуля
«Програмування мовою Асемблер»*

Прийняв
доцент кафедри ТК
Лісовиченко О. І.
“ ... ” 2020 р.

Виконав
студент групи ІІІ-91
Кінчур В. В.

Київ 2020

Комп'ютерний практикум №1

Тема: Створення програм на асемблері

Завдання: 1. Для програми, наведеної в умові лабораторної, створити файл типу .asm. Ця програма не має засобів виводу даних, тому правильність її виконання треба перевірити за допомогою виконуваного файлу та налагоджувача.

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 .o-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 gdb одержаний виконуваний файл програми.

7. За допомогою консолі gdb знайти початкову адресу сегмента даних та записати його до звіту. Знайти масиви SOURCE та DEST. Дані у масиві SOURCE подаються у десятковій системі.

8. У покроковому режимі за допомогою команди si виконати програму.

Одержані результати у масиві DEST показати викладачеві.

Текст програми (NASM):

```
section .text
```

```
    global _start
```

```
_start:
```

```
    ; nullify al
```

```
    xor al, al
```

```
    ; nullify array
```

```
    mov [dest], al
```

```
    mov [dest+1], al
```

```
    mov [dest+2], al
```

```
    mov [dest+3], al
```

```
    ; OR : mov byte [dest+i], 0
```

```
    ; readdressing
```

```
    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
```

```
    ; sys_exit
```

```
    mov eax, 1
```

```
mov ebx, 0
int 80h
```

```
section .data
```

```
source db 10, 20, 30, 40
dest times 4 db "?"
```

Введені та отримані результати:

Вміст .lst файлу:

```
1                section .text
2                                global _start
3
4                _start:
5                                ; nullify al
6 00000000 30C0                                xor al, al
7
8                                ; nullify array
9 00000002 A2[04000000]                mov [dest], al
10 00000007 A2[05000000]                mov [dest+1], al
11 0000000C A2[06000000]                mov [dest+2], al
12 00000011 A2[07000000]                mov [dest+3], al
13
14                                ; OR : mov byte [dest+i], 0
15
16                                ; readdressing
17 00000016 A0[00000000]                mov al, [source]
18 0000001B A2[07000000]                mov [dest+3], al
```

19 00000020 A0[01000000]	mov al, [source+1]
20 00000025 A2[06000000]	mov [dest+2], al
21 0000002A A0[02000000]	mov al, [source+2]
22 0000002F A2[05000000]	mov [dest+1], al
23 00000034 A0[03000000]	mov al, [source+3]
24 00000039 A2[04000000]	mov [dest], al
25	
26	; sys_exit
27 0000003E B801000000	mov eax, 1
28 00000043 BB00000000	mov ebx, 0
29 00000048 CD80	int 80h
30	
31	
32	section .data
33 00000000 0A141E28	source db 10, 20, 30, 40
34 00000004 3F<rept>	dest times 4 db "?"
35	

Вміст .map файлу:

- NASM Map file -----

Source file: lab1/lab1.asm

Output file: lab1.com

-- Program origin -----

00000000

-- Sections (summary) -----

Vstart	Start	Stop	Length	Class Name
	0	0	34 00000034	progbits .text
34	34		3C 00000008	progbits .data

-- Sections (detailed) -----

---- Section .text -----

class: progbits

length: 34

start: 0

align: not defined

follows: not defined

vstart: 0

valign: not defined

vfollows: not defined

---- Section .data -----

class: progbits

length: 8

start: 34

align: 4

follows: not defined

vstart: 34

valign: not defined

vfollows: not defined

-- Symbols -----

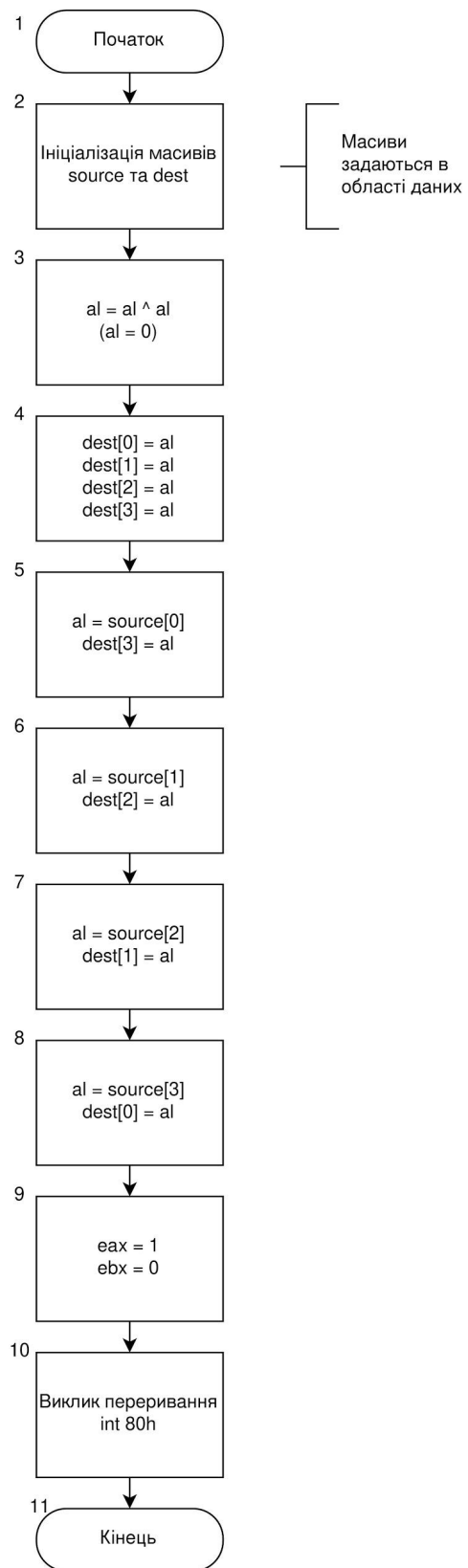
---- Section .text -----

Real	Virtual	Name
	0	0 _start

---- Section .data -----

Real	Virtual	Name
34	34	source
38	38	dest

Схема функціонування програми:



До виконання програми

Масив source:

```
(gdb) x /4d &source  
0x804a000 <source>: 10 20 30 40
```

Масив dest:

```
(gdb) x /4d &dest  
0x804a004 <dest>: 63 63 63 63
```

Після виконання програми

Масив dest:

```
(gdb) x /4d &dest  
0x804a004 <dest>: 40 30 20 10
```

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 .o-файл програми, включивши опції для налагодження та створення .map-файлу.
 5. Відкрив файл карти пам'яті (.map-файл) та подивився на адреси початку та кінця всіх сегментів програми.
 6. Завантажив програму налагоджувача gdb та мій одержаний виконуваний файл програми.
 7. В сегменті даних знайшов масиви SOURCE та DEST та вивів відповідні значення у десятковій системі.
 8. У покроковому режимі за допомогою команди si виконав програму.
- Програма коректно виконує поставлену задачу.