

Лабораторная работа №7

Архитектура компьютера

Ли Евгения Олеговна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	создала файл lab7-1.asm	8
4.2	результат символ j.	8
4.3	Исправила текст программы следующим образом:	9
4.4	Вывелся символ с кодом 10	10
4.5	Файл lab7-2.asm	10
4.6	Получила число 106.	10
4.7	Изменение	10
4.8	Результат 10	11
4.9	iprintLF на iprint	11
4.10	Результат	11
4.11	Создала исполняемый файл и запустила его.	12
4.12	Текст	12
4.13	Создала исполняемый файл и запустила его	12
4.14	Создала исполняемый файл и запустила его.	13
4.15	Программа	14
4.16	Проверила его работу для значений x=1 и x=4	14

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2 Задание

Освоить арифметические инструкции языка ассемблера NASM

3 Теоретическое введение

Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: `mov ax,bx`.

Непосредственная адресация – значение операнда задается непосредственно в команде, Например: `mov ax,2`

4 Выполнение лабораторной работы

1. Создала каталог для программ лабораторной работы No 7, перешла в него и создала файл lab7-1.asm:(рис. 4.1)

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ touch lab7-1.asm
```

Рис. 4.1: создала файл lab7-1.asm

2. Ввела в файл lab7-1.asm текст программы из листинга 7.1

Создала исполняемый файл и запустила его.(рис. 4.2)

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-1.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-1
j
```

Рис. 4.2: результат символ j.

3. Далее изменила текст программы и вместо символов, записала в регистры числа. (рис. 4.3)



```
lab7-1.asm [-M--]
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
mov [buf1],eax
mov eax,buf1
call sprintLF
call quit
```

Рис. 4.3: Исправила текст программы следующим образом:

Создала исполняемый файл и запустила его.(рис. 4.4)

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-1.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-1

eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-1
```

Рис. 4.4: Вывелся символ с кодом 10

Символ при выводе на экран не отображается

4. Создала файл lab7-2.asm и ввела в него текст программы из листинга 7.2.(рис. 4.5, 4.6)

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ touch lab7-2.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $
```

Рис. 4.5: Файл lab7-2.asm

Создала исполняемый файл и запустила его

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-2.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-2
106
```

Рис. 4.6: Получила число 106.

5. Изменила символы на числа. (рис. 4.7, 4.8, 4.9, 4.10)

```
lab7-2.asm [-M--] 9 L: [ 1+ 5 6/ 10] *(77 / 114b) 0010 0x00A
#include "in_out.asm"
SECTION text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Подтверждаете запись файла "/afs/.dk

Рис. 4.7: Изменение

Создала исполняемый файл и запустила его.

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ mc
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-2.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-2
10
```

Рис. 4.8: Результат 10

Заменяла функцию `iprintLF` на `iprint`.

```
lab7-2.asm [-M--] 11 L:[ 1-
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit
```

Рис. 4.9: `iprintLF` на `iprint`

```
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ mc
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-2.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-2
10
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $
```

Рис. 4.10: Результат

7.3.2. Выполнение арифметических операций в NASM

6. Создала файл `lab7-3.asm`. Ввела текст программы из листинга 7.3 в `lab7-3.asm`. (рис. 4.11)

```

eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-3.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-3
Результат: 4
Остаток от деления: 1
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $

```

Рис. 4.11: Создала исполняемый файл и запустила его.

Изменила текст программы для вычисления выражения $y=(4*6+2)/5$ (рис. 4.12, 4.13)

```

lab7-3.asm      [----]  0 L: [ 1+29 30/ 30] *(1366/1366b) <EOF>
;
; Программа вычисления выражения
;
%include "in_out.asm" ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=4
mov ebx,6 ; EBX=6
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; EBX=5
div ebx ; EAX=EAX/5, EDX=остаток от деления
mov edi,eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax,div ; вызов подпрограммы печати
call sprint ; сообщения 'Результат: '
mov eax,edi ; вызов подпрограммы печати значения
call iprintLF ; из 'edi' в виде символов
mov eax,rem ; вызов подпрограммы печати
call sprint ; сообщения 'Остаток от деления: '
mov eax,edx ; вызов подпрограммы печати значения
call iprintLF ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 4.12: Текст

```

eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf lab7-3.asm
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./lab7-3
Результат: 5
Остаток от деления: 1
eoli@dk8n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $

```

Рис. 4.13: Создала исполняемый файл и запустила его

7. Создала файл variant.asm. Ввела текст программы из листинга 7.4 в файл

variant.asm.(рис. 4.14)

```
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf variant.asm
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o variant variant.o
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ mc
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./variant
Введите No студенческого билета:
1132226492
Ваш вариант: 13
```

Рис. 4.14: Создала исполняемый файл и запустила его.

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

```
mov eax, msg call sprintLF
```

2. Для чего используются следующие инструкции? `nasm mov ecx, xmov edx, 80 call sread`

Для ввода переменной X с клавиатуры и сохранения введенных данных

3. Для чего используется инструкция “`call atoi`”?

Для преобразования кода переменной ASCII в число

4. Какие строки листинга 7.4 отвечают за вычисления варианта?

```
mov ebx, 20 div ebx, inc edx
```

5. В какой регистр записывается остаток от деления при выполнении инструкции “`div ebx`”?

ebx

6. Для чего используется инструкция “`inc edx`”?

Для увеличения значения edx на 1

7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычисления?

mov eax, edx call iprint LF

7.4. Задание для самостоятельной работы

1. Написала программу вычисления выражения $y=(8x+6)10$. рис. 4.15, 4.16)

```
evgenia.asm [-M--] 9 L: [ 1+15 16/ 40] *(287 / 503b) 0109 0x06D
%include "in_out.asm"
SECTION .data
rem: DB "Вычислить значение выражения (8x+6)10",0
msg: DB "Введите x: ",0
div: DB "Результат: ",0

SECTION .bss
x: RESB 80
rez: RESB 80
SECTION .text
GLOBAL _start
_start:

mov eax, rem
call sprintf
mov eax, msg
call sprintf

mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi

mov ebx, 8
mul ebx
add eax, 6
xor edx, edx
mov ecx, 10
div ecx

mov [rez], eax
mov eax, div
call sprintf
mov eax, [rez]
call iprintLF
```

Рис. 4.15: Программа

```
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ nasm -f elf evgenia.asm
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ld -m elf_i386 -o evgenia evgenia.o
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./evgenia
Вычислить значение выражения (8x+6)10
Введите x:
1
Результат:
1
eoli@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab07 $ ./evgenia
Вычислить значение выражения (8x+6)10
Введите x:
4
Результат:
3
```

Рис. 4.16: Проверила его работу для значений $x=1$ и $x=4$

5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM

Список литературы