

Лабораторная работа №7.Арифметические Операции в NASM

Архитектура ЭВМ

Алексей Назаров НММбд-02-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Складывание строк	6
2.2	Складывание чисел	7
2.3	Программа вывода значение регистра eax	8
2.3.1	Создадим файл lab7-2.asm	8
2.3.2	Заменим sprintf на fprintf	9
2.4	Программа для вычисления выражения $f(x)$	9
2.5	Вычисления выражения $f(x) = (4*6 + 2)/5$	10
2.6	Вычисление варианта по студенческому билету	11
3	Задания для самостоятельной работы	13
4	Выводы	15

Список иллюстраций

2.1	Создание lab-1.asm	6
2.2	Текст в lab-1.asm	7
2.3	Компиляция и запуск кода	7
2.4	Выполнение кода	7
2.5	Создание файла lab7-2.asm	8
2.6	Код lab7-2.asm	8
2.7	Компиляция и запуск	9
2.8	Запуск измененной программы	9
2.9	Запуск ./lab7-2	9
2.10	Создание файла lab7-3.asm	9
2.11	Компиляция и запуск lab7-3.asm	10
2.12	Изменения кода для вычисления другой функции	10
2.13	Вывод lab7-3.asm	11
2.14	Код variant.asm	11
2.15	Запуск программы по вычислению варианта	12
3.1	Код, вычисления выражения $(8x+6) * 10$	13
3.2	Результаты для 1 и 4	14

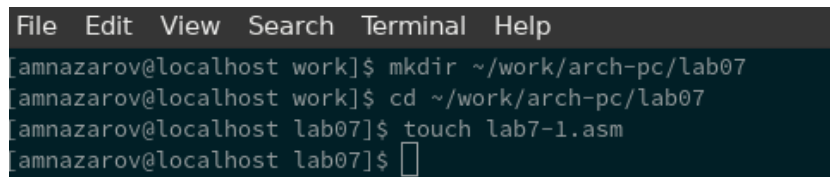
Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Выполнение лабораторной работы

Создадим каталог и перейдем в него. Создадим lab7-1.asm



```
File Edit View Search Terminal Help
[amnazarov@localhost work]$ mkdir ~/work/arch-pc/lab07
[amnazarov@localhost work]$ cd ~/work/arch-pc/lab07
[amnazarov@localhost lab07]$ touch lab7-1.asm
[amnazarov@localhost lab07]$
```

Рис. 2.1: Создание lab-1.asm

2.1 Складывание строк

Введем текст листинга в файл

```

1 %include 'in_out.asm'
2 SECTION .bss
3
4 buf1: RESB 80
5     SECTION .text
6     GLOBAL _start
7
8
9 _start:
10 mov eax,'6'
11 mov ebx,'4'
12 add eax,ebx
13 mov [buf1],eax
14 mov eax,buf1
15 call sprintLF
16
17 call quit
18

```

Рис. 2.2: Текст в lab-1.asm

Оттранслируем, слинкуем и запустим

```

[amnazarov@localhost lab07]$ nasm -f elf lab7-1.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[amnazarov@localhost lab07]$ ./lab7-1
j
[amnazarov@localhost lab07]$

```

Рис. 2.3: Компиляция и запуск кода

Вывелось “j”

2.2 Складывание чисел

Уберем кавычки и скомпилируем

```

[amnazarov@localhost lab07]$ nasm -f elf lab7-1.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[amnazarov@localhost lab07]$ ./lab7-1

```

Рис. 2.4: Выполнение кода

Вывелся перенос строки (его символ 10)

2.3 Программа вывода значение регистра eax

2.3.1 Создадим файл lab7-2.asm

```
[amnazarov@localhost lab07]$ touch lab7-2.asm  
[amnazarov@localhost lab07]$ ls  
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2.asm  
[amnazarov@localhost lab07]$
```

Рис. 2.5: Создание файла lab7-2.asm

Введем код

```
1 %include 'in_out.asm'  
2 SECTION .text  
3 GLOBAL _start  
4  
5 _start:  
6  
7 mov eax, '6'  
8  
9 mov ebx, '4'  
10 add eax, ebx  
11 call iprintLF  
12 call quit
```

Рис. 2.6: Код lab7-2.asm

Скомпилируем и запустим


```
[amnazarov@localhost lab07]$ nasm -f elf lab7-2.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[amnazarov@localhost lab07]$ ./lab7-2
106
[amnazarov@localhost lab07]$
```

Рис. 2.7: Компиляция и запуск

Программа вывела 106, так как мы суммируем адреса двух строк

2.3.2 Заменим `sprintLF` на `iprintLF`

```
[amnazarov@localhost lab07]$ nasm -f elf lab7-2.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[amnazarov@localhost lab07]$ ./lab7-2
10
[amnazarov@localhost lab07]$
```

Рис. 2.8: Запуск измененной программы

Вывелось 10

Заменим `iprintLF` на `iprint`

```
[amnazarov@localhost lab07]$ ./lab7-2
10[amnazarov@localhost lab07]$
```

Рис. 2.9: Запуск `./lab7-2`

Вывелось 10, без перевода строки

2.4 Программа для вычисления выражения $f(x)$

Создадим файл `lab7-3.asm`

```
[amnazarov@localhost lab07]$ touch lab7-3.asm
[amnazarov@localhost lab07]$ ^C
[amnazarov@localhost lab07]$ ls
in_out.asm  lab7-1.asm  lab7-2      lab7-2.o
lab7-1      lab7-1.o    lab7-2.asm  lab7-3.asm
[amnazarov@localhost lab07]$
```

Рис. 2.10: Создание файла `lab7-3.asm`

Скомпилируем и запустим Код.

```
[amnazarov@localhost lab07]$ nasm -f elf lab7-3.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[amnazarov@localhost lab07]$ ./lab7-3
Результат: 4
Остаток от деления: 1
[amnazarov@localhost lab07]$
```

Рис. 2.11: Компиляция и запуск lab7-3.asm

2.5 Вычисления выражения $f(x) = (4*6 + 2)/5$

Заменяем числа на 4, 6, 2 и 5

```
1 %include 'in_out.asm' ; подключение внешнего файла
2
3 SECTION .data
4 div: DB 'Результат: ',0
5 rem: DB 'Остаток от деления: ',0
6
7 SECTION .text
8 GLOBAL _start
9 _start: ; ---- Вычисление выражения
10 mov eax,4 ; EAX=5
11 mov ebx,6 ; EBX=2
12 mul ebx ; EAX=EAX*EBX
13
14
15 add eax,2 ; EAX=EAX+3
16 xor edx,edx ; обнуляем EDX для корректной работы div
17 mov ebx,5 ; EBX=3
18 div ebx ; EAX=EAX/3, EDX=остаток от деления
19
20 mov edi,eax ; запись результата вычисления в 'edi'
21
22 ; ---- Вывод результата на экран
23
24 mov eax,div ; вызов подпрограммы печати
25 call sprint ; сообщения 'Результат: '
26 mov eax,edi ; вызов подпрограммы печати значения
27 call iprintLF ; из 'edi' в виде символов
28
29 mov eax,rem ; вызов подпрограммы печати
30 call sprint ; сообщения 'Остаток от деления: '
31 mov eax,edx ; вызов подпрограммы печати значения
32 call iprintLF ; из 'edx' (остаток) в виде символов
33
34 call quit ; вызов подпрограммы завершения
35
36
```

Рис. 2.12: Изменения кода для вычисления другой функции

Скомпилируем и запустим

```

[amnazarov@localhost lab07]$ nasm -f elf lab7-3.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7-3 lab7-3.o
[amnazarov@localhost lab07]$ ./lab7-3
Результат: 5
Остаток от деления: 1
[amnazarov@localhost lab07]$

```

Рис. 2.13: Вывод lab7-3.asm

2.6 Вычисление варианта по студенческому билету

Создадим файл `variant.asm` с текстом листинга 7.4

```

1 ;-----
2 ; Программа вычисления варианта
3 ;-----
4 %include 'in_out.asm'
5
6 SECTION .data
7 msg: DB 'Введите № студенческого билета: ',0
8 rem: DB 'Ваш вариант: ',0
9
10 SECTION .bss
11 x:    RESB 80
12
13 SECTION .text
14 GLOBAL _start
15 _start:
16     mov eax, msg
17     call sprintf
18
19     mov ecx, x
20     mov edx, 80
21     call sread
22
23     mov eax, x ; вызов подпрограммы преобразования
24     call atoi ; ASCII кода в число, `eax=x`
25
26     xor edx, edx
27     mov ebx, 20
28     div ebx
29     inc edx
30
31     mov eax, rem
32     call sprintf
33     mov eax, edx
34     call iprintLF
35
36     call quit
37

```

Рис. 2.14: Код variant.asm

1. Для вывода “Ваш вариант” используются строки 31-32

2. инструкции на строка 19-21 используется для ввода текста в `x` из консоли
3. `call atoi` используется для преобразования ASCII кода в число
4. За вычисление варианта отвечают строки 26-29
5. Остаток от деления записывается в `edx`
6. `inc edx` увеличивает значение `edx` на 1
7. За вывод результата вычислений отвечают строки 33-34 (выделенно желтым)

Запустим программу и узнаем номер варианта

```
[amnazarov@localhost lab07]$ nasm -f elf variant.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o variant variant.o
[amnazarov@localhost lab07]$ ./variant
Введите № студенческого билета:
1132226512
Ваш вариант: 13
[amnazarov@localhost lab07]$
```

Рис. 2.15: Запуск программы по вычислению варианта

Получили 13

3 Задания для самостоятельной работы

Написать программу для вычисления $(8x+6)*10$

Напишем, код, который будет вычислять это выражение

```
1 ;-----
2 ; Программа вычисления варианта
3 ;-----
4 %include 'in_out.asm'
5
6 SECTION .data
7 msg: DB 'x: ',0
8 rem: DB 'Ответ: ',0
9
10 SECTION .bss
11 x:    RESB 80
12
13 SECTION .text
14 GLOBAL _start
15 _start:
16     mov eax, msg
17     call sprintLF
18
19     mov ecx, x
20     mov edx, 80
21     call sread
22
23     mov eax, x ; вызов подпрограммы преобразования
24     call atoi ; ASCII кода в число, `eax=x`
25
26     mov edx, 8
27     mul edx
28     mov edx, 6
29     add eax, edx
30     mov edx, 10
31     mul edx
32     mov edx, eax
33
34     mov eax, rem
35     call sprint
36     mov eax, edx
37     call iprintLF
38
39     call quit
40
```

Рис. 3.1: Код, вычисления выражения $(8x+6) * 10$

Запустим код и введем туда 1 и 4

```
[amnazarov@localhost lab07]$ nasm -f elf lab7.asm
[amnazarov@localhost lab07]$ ld -m elf_i386 -o lab7 lab7.o
[amnazarov@localhost lab07]$ ./lab7
x:
1
Ответ: 140
[amnazarov@localhost lab07]$ ./lab7
x:
4
Ответ: 380
[amnazarov@localhost lab07]$
```

Рис. 3.2: Результаты для 1 и 4

4 Выводы

Мы освоили арифметические операции в NASM