

Лабораторная работа №6

Арифметические операции в NASM

Колонтырский Илья Русланович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	16

Список иллюстраций

2.1	Создание файла и рабочей директории	6
2.2	Запись кода из листинга	6
2.3	Сборка исполняемого файла и запуск	7
2.4	Редактирование файла	7
2.5	Запуск программы и сборка	7
2.6	Создание файла lab6-2.asm	8
2.7	Вставка кода в файл lab6-2.asm	8
2.8	Запуск файла и результат	8
2.9	Изменение файла	9
2.10	Сборка файла и результат	9
2.11	Редактирование файла	9
2.12	Сборка и результат работы отредактированного файла	10
2.13	Создание файла lab6-3.asm	10
2.14	Вставка кода в файл	10
2.15	Сборка и запуск	11
2.16	Изменение файла	11
2.17	Повторная сборка и запуск	12
2.18	Создание файла variant.asm	12
2.19	Вставка кода	12
2.20	Компиляция и запуск	13
2.21	Создание файла	14
2.22	Код программы	14
2.23	Запуск программы и проверка её корректной работы	15

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

2 Выполнение лабораторной работы

Создадим папку рабочего каталога и файл lab6-1.asm (Рис. 2.1)

```
irkolontyrskiy@irkolontyrskiy:~$ mkdir ~/work/arch-pc/lab06
irkolontyrskiy@irkolontyrskiy:~$ cd ~/work/arch-pc/lab06
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ touch lab6-1.asm
```

Рис. 2.1: Создание файла и рабочей директории

Вставим в созданный файл код из листинга (Рис. 2.2)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-1.asm *
#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. 2.2: Запись кода из листинга

Теперь соберём файл и запустим его (Рис. 2.3)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-1
j
```

Рис. 2.3: Сборка исполняемого файла и запуск

Вывело символ j. Изменим файл (Рис. 2.4)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-1.asm
#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. 2.4: Редактирование файла

Мы убрали кавычки и теперь складываем числа. Соберём и запустим файл (Рис. 2.5)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-1
```

Рис. 2.5: Запуск программы и сборка

Ничего не вывелось. Но на самом деле вывелся символ перевода строки, и поэтому мы его не видим. Создадим файл под названием lab6-2.asm (Рис. 2.6)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm
```

Рис. 2.6: Создание файла lab6-2.asm

Вставим в него следующий код (Рис. 2.7)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-2.asm *
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit
```

Рис. 2.7: Вставка кода в файл lab6-2.asm

Вместо sprintLF тут iprintLF. Соберём файл и запустим (Рис. 2.8)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-2
106
```

Рис. 2.8: Запуск файла и результат

Вывело число 106. Уберём кавычки у чисел в коде (Рис. 2.9)


```

GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-2.asm *
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit

```

Рис. 2.9: Изменение файла

Соберём и запустим программу (Рис. 2.10)

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab
6-2.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-2
10

```

Рис. 2.10: Сборка файла и результат

Вывод корректный. Заменим iprintLF на iprint (Рис. 2.11)

```

GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-2.asm *
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

```

Рис. 2.11: Редактирование файла

Соберём и запустим (Рис. 2.12)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-2
10irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$
```

Рис. 2.12: Сборка и результат работы отредактированного файла

iprint не использует перенос на новую строку. Создадим ещё 1 файл (Рис. 2.13)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
```

Рис. 2.13: Создание файла lab6-3.asm

Он считает выражение $(5*2+3)/3$. Вставим в него следующий код (Рис. 2.14)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-3.asm *
#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,5 ; EAX=5
mov ebx,2 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,3 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,3 ; EBX=3
div ebx ; EAX=EAX/3, 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
```

Рис. 2.14: Вставка кода в файл

Попробуем запустить эту программу, предварительно её собрав (Рис. 2.15)

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 2.15: Сборка и запуск

Вычисления верные. Сделаем так, чтобы код вычислял функцию $f(x)=(4*6+2)/5$. Внесём следующие изменения (Рис. 2.16)

```

GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/lab6-3.asm *
#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

```

Рис. 2.16: Изменение файла

Соберём и запустим (Рис. 2.17)

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

Рис. 2.17: Повторная сборка и запуск

Ответ верный. Создадим файл variant.asm. Он должен вычислять номер варианта по студенческому билету (Рис. 2.18)

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm

```

Рис. 2.18: Создание файла variant.asm

Вставим в него следующий код (Рис. 2.19)

```

GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/variant.asm *
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx, edx
mov ebx, 20
div ebx
inc edx

```

Рис. 2.19: Вставка кода

Скомпилируем и запустим файл. Укажем в вводе студенческий билет (Рис. 2.20):

```

irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf variant.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant va
riant.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132237378
Ваш вариант: 19

```

Рис. 2.20: Компиляция и запуск

Пересчитав вручную, находим, что это верный ответ

Ответим на вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

Строчка под номером 25

2. Для чего используются следующие инструкции?

`mov ecx, x`

`mov edx, 80`

`call sread`

Они записывают ввод с клавиатуры в переменную `x`

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

Для преобразования кода символа в число

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

`div ebx`

`inc edx`

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

В `edx`

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

Прибавляет к `edx` единицу

7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычисления?

`mov eax,edx`

call iprintLF

Самостоятельная работа

Создадим файл самостоятельной работы (Рис. 2.21)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ touch var19.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$
```

Рис. 2.21: Создание файла

Теперь напишем следующий код (Рис. 2.22)

```
GNU nano 6.2 /home/irkolontyrskiy/work/arch-pc/lab06/var19.asm
#include 'in_out.asm'
SECTION .data
msg1: DB 'f(x)=((1/3)x+5)*7',0
msg2: DB 'Введите X: ',0
rem: DB 'Ответ: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg1
call sprintLF
mov eax, msg2
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, `eax=x`
xor edx, edx
mov ebx, 3
div ebx
add eax, 5
mov ebx, 7
mul ebx
mov edx, eax
mov eax, rem
call sprint
mov eax, edx
call iprintLF
call quit
```

Рис. 2.22: Код программы

Проверим работу программы, собрав и запустив её (Рис. 2.23)

```
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ nasm -f elf var19.asm
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ld -m elf_i386 -o var19 var19.o
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./var19
f(x)=((1/3)x+5)*7
Введите X:
3
Ответ: 42
irkolontyrskiy@irkolontyrskiy:~/work/arch-pc/lab06$ ./var19
f(x)=((1/3)x+5)*7
Введите X:
9
Ответ: 56
```

Рис. 2.23: Запуск программы и проверка её корректной работы

Программа дала верные ответы

3 Выводы

Были получены навыки работы с арифметическими инструкциями языка ассемблера