

Лабораторная работа №6

Арифметические операции в NASM

Глобин Никита Анатольевич

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 3.1 | Символьные и численные данные в NASM | 7 |
| 3.2 | Выполнение арифметических операций в NASM | 10 |
| 3.3 | Ответы на вопросы | 13 |
| 3.4 | Задание для самостоятельной работы | 15 |
| 3.5 | Выводы | 16 |
| | Список литературы | 17 |

Список иллюстраций

| | | |
|------|----------|----|
| 3.1 | photo 1 | 7 |
| 3.2 | photo 2 | 7 |
| 3.3 | photo 3 | 7 |
| 3.4 | photo 4 | 8 |
| 3.5 | photo 5 | 8 |
| 3.6 | photo 6 | 9 |
| 3.7 | photo 7 | 9 |
| 3.8 | photo 8 | 9 |
| 3.9 | photo 9 | 10 |
| 3.10 | photo 10 | 10 |
| 3.11 | photo 11 | 10 |
| 3.12 | photo 12 | 11 |
| 3.13 | photo 13 | 11 |
| 3.14 | photo 14 | 11 |
| 3.15 | photo 15 | 12 |
| 3.16 | photo 16 | 12 |
| 3.17 | photo 17 | 13 |
| 3.18 | photo 18 | 13 |
| 3.19 | photo 19 | 15 |
| 3.20 | photo 20 | 15 |
| 3.21 | photo 21 | 15 |

Список таблиц

1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Символьные и численные данные в NASM

Выполнение арифметических операций в NASM

Ответы на вопросы

Задание для самостоятельной работы

3 Выполнение лабораторной работы

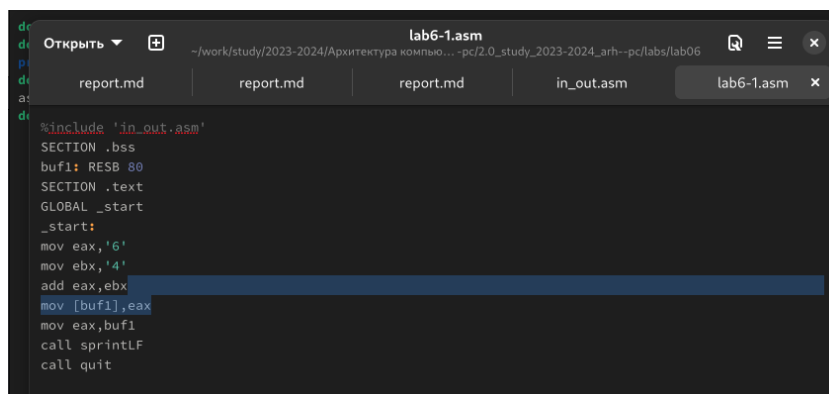
3.1 Символьные и численные данные в NASM

1. переходим в каталог lab06 и создаём там файл lab6-1.asm (рис. 3.1).

```
dodo@vbox:~$ cd work/study/2023-2024/Архитектура\ компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ls
presentation report
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ touch lab6-1.
asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.1: photo 1

переписываем код программы из ТУИС (рис. 3.2).



```
lab6-1.asm
~/work/study/2023-2024/Архитектура компью... -pc/2.0_study_2023-2024_arh--pc/labs/lab06
report.md report.md report.md in_out.asm lab6-1.asm x
%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 sprintf
call quit
```

Рис. 3.2: photo 2

компилируем и запускаем (рис. 3.3).

```
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf lab6-1.asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-1
j
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.3: photo 3

2. Далее изменим текст программы и вместо символов, запишем в регистры числа. (рис. 3.4).

```
%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. 3.4: photo 4

компилируем и запускаем (рис. 3.5).

```
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf lab6-1.asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-1

dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.5: photo 5

3. создаём новый файл lab6-2 (рис. 3.6).


```
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ touch lab6-2.asm
```

Рис. 3.6: photo 6

переписываем код программы из ТУИС (рис. 3.7).

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax, '6|'
mov ebx, '4'
add eax, ebx
call iprintLF
call quit
```

Рис. 3.7: photo 7

компилируем и запускаем (рис. 3.8).

```
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf lab6-2.asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-2
106
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.8: photo 8

4. Аналогично предыдущему примеру изменим символы на числа (рис. 3.9).

```
%include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit
```

Рис. 3.9: photo 9

компилируем и запускаем (рис. 3.10).

```
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-2
10
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.10: photo 10

5. Заменяем функцию iprintLF на iprint и выведи это (рис. 3.11).

```
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf lab6-2.asm
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-2
10dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.11: photo 11

теперь ответ не пишется в отдельной строке.

3.2 Выполнение арифметических операций в NASM

1. Создаём файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. 3.12).

```

dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ touch lab6-
10dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$

```

Рис. 3.12: photo 12

2. переписываем код программы из ТУИС (рис. 3.13).

```

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

```

Рис. 3.13: photo 13

3. компилируем и запускаем (рис. 3.14).

```

dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf lab6-3.asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

Рис. 3.14: photo 14

4. Изменяем текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (рис. 3.15).

```

SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax,4 ; EAX=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+3
xor edx,edx ; обнуляем EDX для корректной работы div
mov ebx,5 ; 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
call iprintf ; 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 iprintf ; из 'edx' (остаток) в виде символов
call quit ; вызов подпрограммы завершения

```

Рис. 3.15: photo 15

5. компилируем и запускаем (рис. 3.16).

```

dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf lab6-3.asm
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$

```

Рис. 3.16: photo 16

6. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму (рис. 3.17).

```

;-----
; Программа вычисления варианта
;-----
#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
mov eax, rem
call sprintf
mov eax, edx
call iprintf
call quit

```

Рис. 3.17: photo 17

7. компилируем и запускаем (рис. 3.18).

```

dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./variant
Введите № студенческого билета:
1234567808
Ваш вариант: 9
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$

```

Рис. 3.18: photo 18

3.3 Ответы на вопросы

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

mov eax, rem ; Загрузка адреса сообщения “Ваш вариант:” в регистр EAX
call sprintf ; Вызов подпрограммы для вывода сообщения

2. Для чего используются следующие инструкции? `mov esx, x` Устанавливает указатель на буфер `x`, куда будет записан введенный текст.

`mov edx, 80` Указывает максимальную длину ввода (80 байт).

`call sread` Вызывает подпрограмму для считывания строки из ввода (с клавиатуры) и записи её в буфер `x`.

3. Для чего используется инструкция `call atoi`? Она преобразует строку (символы ASCII) из буфера `x` в число и записывает результат в регистр `eax`. Это необходимо для выполнения арифметических операций с введенными данными.

4. Какие строки листинга 6.4 отвечают за вычисление варианта? Строки:

`xor edx, edx` ; Обнуление регистра EDX `mov ebx, 20` ; Загрузка делителя (20) в EBX
`div ebx` ; Деление: результат в EAX, остаток в EDX `inc edx` ; Увеличение остатка на 1 для получения номера варианта

5. В какой регистр записывается остаток от деления при выполнении инструкции `div ebx`? Остаток от деления записывается в регистр EDX.

6. Для чего используется инструкция `inc edx`? Инструкция `inc edx` увеличивает остаток от деления на единицу. Это необходимо, чтобы результат вычисления соответствовал формуле:

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

`mov eax, rem` ; Загрузка адреса сообщения "Ваш вариант:" в EAX `call sprint` ; Вывод сообщения `mov eax, edx` ; Загрузка номера варианта из EDX в EAX `call iprintLF` ; Вывод номера варианта с переводом строки

3.4 Задание для самостоятельной работы

1. создаём файл (рис. 3.19).

```
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ touch task9.asm
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.19: photo 19

2. пишем в нём код (рис. 3.20).

```

;-----
; Программа вычисления варианта
;-----
%include 'in_out.asm'
SECTION .data
msg: DB 'Выражение для вычисления: 10+(31x-5)',0
msg2: DB 'Введите X: ',0
SECTION .bss
x: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintf
mov eax, msg2
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'
mov ebx, 31
mul ebx
sub eax, 5
add eax, 10
call iprintf
call quit
```

Рис. 3.20: photo 20

3. компилируем и запускаем (рис. 3.21).

```
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ touch task9.asm
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ nasm -f elf task9.asm
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ld -m elf_i386 -o task9 task9.o
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$ ./task9
Выражение для вычисления: 10+(31x-5)
Введите X:
4
129
dodo@vbox: ~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/labs/lab06$
```

Рис. 3.21: photo 21

3.5 Выводы

В результате выполнения лабораторной работы были достигнуты следующие результаты, соответствующие поставленной цели:

1. Освоение арифметических инструкций NASM: Были изучены основные команды языка ассемблера для выполнения арифметических операций: сложение (add), вычитание (sub), умножение (mul и imul), деление (div и idiv), а также инкремент и декремент (inc, dec).
2. Работа с числовыми и символьными данными: Изучены методы преобразования данных между символьным и числовым представлением с использованием ASCII-кодов и специальных подпрограмм (atoi, iprint и др.). Это позволило корректно вводить данные с клавиатуры и выводить результаты на экран.
3. Практическое применение арифметических операций: Было успешно реализовано вычисление сложных арифметических выражений на языке NASM с учётом правил адресации данных и особенностей использования регистров.
4. Закрепление навыков программирования на ассемблере: Выполнение лабораторной работы позволило на практике закрепить теоретические знания по архитектуре ЭВМ и программированию на языке ассемблера NASM.

Список литературы