

# Отчет по выполнению лабораторной работы №6

## Дисциплина: архитектура компьютеров

Новиков Никита Владимирович

### Содержание

#### 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

#### 2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM

#### 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти.

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`.

Адресация памяти – операнд задает адрес в памяти. В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

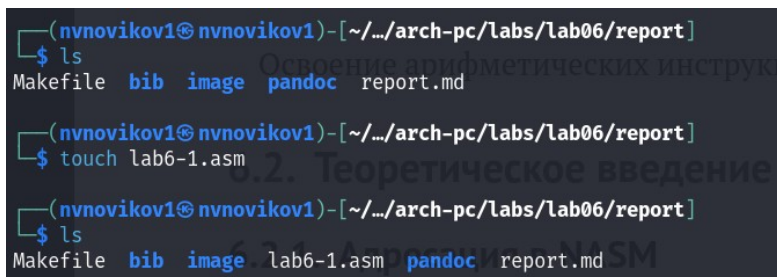
Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, например, чтобы вывести число, надо предварительно преобразовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как

последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Введенные данные будут представлять собой символы, что делает невозможным получение корректного результата при выполнении над ними арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно

## 4 Выполнение лабораторной работы

### 4.1 Символьные и численные данные в NASM

Создаю каталог для программ лабораторной работы №6, перехожу в него и создаю файл lab6-1.asm (рис. [??]).



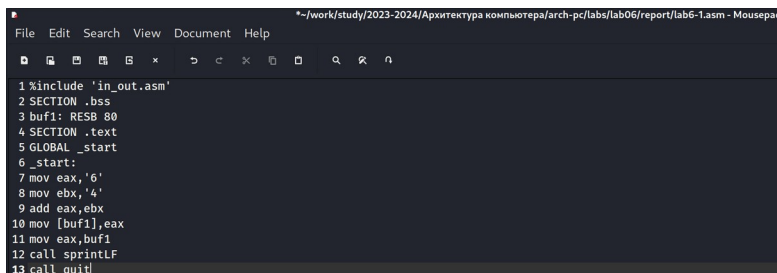
```
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ls
Makefile  bib  image  pandoc  report.md

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ touch lab6-1.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ls
Makefile  bib  image  lab6-1.asm  pandoc  report.md
```

*Создание каталога*

Ввожу в файл lab6-1.asm текст программы из листинга 6.1.(рис. [??]).



```
*~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab06/report/lab6-1.asm - Mousepad
File Edit Search View Document Help
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax,'6'
8 mov ebx,'4'
9 add eax,ebx
10 mov [buf1],eax
11 mov eax,buf1
12 call sprintf
13 call quit
```

*Ввод программы из листинга 6.1*

Создаю исполняемый файл и запускаю его.(рис. [??]).

```

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-1.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-1 lab6-1.o

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ./lab6-1
j

```

### Запуск файла

Изменяю текст программы и вместо символов записываю в регистры числа.  
(рис. [??]).

```

File Edit Search View Document Help
1 %include 'in_out.asm'
2 SECTION .bss
3 buf1: RESB 80
4 SECTION .text
5 GLOBAL _start
6 _start:
7 mov eax, 6
8 mov ebx, 4
9 add eax, ebx
10 mov [buf1], eax
11 mov eax, buf1
12 call sprintf
13 call quit

```

### Исправление программы

Далее создаю исполняемый файл и запускаю его.(рис. [??]).

```

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-1.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-1 lab6-1.o

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ./lab6-1

```

### Запуск исправленной программы

Пользуясь таблицей ASCII можно определить, что код 10 соответствует символу переносу строки. Этот символ не отображается на экране.

Создаю новый файл lab6-2 в том же каталоге и ввожу в него текст программы из листинга 6.2.(рис. [??]).

```

File Edit Search View Document Help
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax, '6'
6 mov ebx, '4'
7 add eax, ebx
8 call iprintf
9 call quit

```

### Создание нового файла

Создаю исполняемый файл и запускаю его.(рис. [??]).

```
(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-2.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-2 lab6-2.o

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ./lab6-2
106
```

### *Запуск новой программы*

Аналогично предыдущему примеру изменяю символы на числа.(рис. [??]).

```
File Edit Search View Document Help
~/.work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab06/report/lab6-2.asm - Mousepad
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprintLF
9 call quit
```

### *Изменение программы*

Создаю исполняемый файл и запускаю его.(рис. [??]).

```
(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-2.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-2 lab6-2.o

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ./lab6-2
10
```

### *Запуск исправленной программы*

При исполнении программы было получено число 10.

Заменяю функцию iprintLF на iprint. Создаю исполняемый файл и запускаю его.(рис. [??]).(рис. [??]).

```
File Edit Search View Document Help
~/.work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab06/report/lab6-2.asm - Mousepad
1 %include 'in_out.asm'
2 SECTION .text
3 GLOBAL _start
4 _start:
5 mov eax,6
6 mov ebx,4
7 add eax,ebx
8 call iprint
9 call quit
```

### *Изменение файла*

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-2.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-2 lab6-2.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ./lab6-2
10

```

## Запуск программы

Вывод функции `iprint` отличается от `iprintLF` тем, что выведенное сообщение не переносится на следующую строку.

## 4.2 Выполнение арифметических операций в NASM

Создаю файл `lab6-3.asm` в каталоге `~/work/arch-pc/lab06`. (рис. [??]).

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ touch lab6-3.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ls
Makefile  bib  image  in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o  lab6-3.asm  pandoc  report.md

```

## Создание файла

После внимательного прочтения текста программы из листинга 6.3 ввожу его в `lab6-3.asm`. (рис. [??]).

```

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

```

## Ввод программы из листинга 6.3

Создаю исполняемый файл и запускаю его. (рис. [??]).

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-3.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-3 lab6-3.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ./lab6-3
Результат: 4
Остаток от деления: 1

```

## Запуск программы

Изменяю тест программы для вычисления выражения  $f(x) = (4*6+2)/5$ . (рис. [??]).

```

File Edit Search View Document Help
[Icons]

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

```

## Редактирование текста программы

Создаю исполняемый файл и запускаю его. (рис. [??]).

```

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-3.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-3 lab6-3.o

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ./lab6-3
Результат: 5
Остаток от деления: 1

```

## Запуск программы

Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06. (рис. [??]).

```

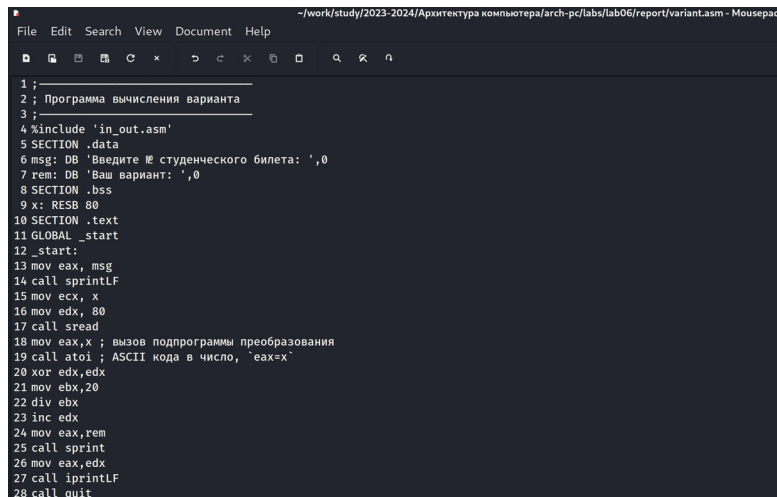
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ touch variant.asm

(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]
$ ls
Makefile  hib  image  in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2  lab6-2.asm  lab6-2.o  lab6-3  lab6-3.asm  lab6-3.o  pandoc  report.md  variant.asm

```

## Создание файла variant.asm

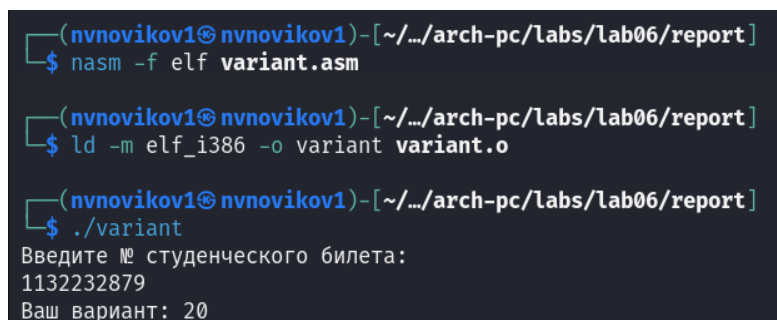
Внимательно изучаю текст программы из листинга 6.4 и ввожу в файл variant.asm.(рис. [??]).



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

Создание файла variant.asm

Создаю исполняемый файл и запускаю его.(рис. [??]).



```
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]  
$ nasm -f elf variant.asm  
  
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]  
$ ld -m elf_i386 -o variant variant.o  
  
(nvnovikov1@nvnovikov1)-[~/.../arch-pc/labs/lab06/report]  
$ ./variant  
Введите № студенческого билета:  
1132232879  
Ваш вариант: 20
```

Запуск файла

Ответы на вопросы

1. За вывод сообщение “Ваш вариант” отвечают строки: mov eax,rem call sprint
2. Эти строки используются чтобы считать x.
3. Call atoi преобразовывает код ASCII в целое число
4. За вычисление варианта отвечают строки: xor edx,edx mov ebx,20 div ebx inc edx
5. Остаток от деления записывается в регистр edx.
6. Инструкция inc edx используется для того, чтобы увеличить значение регистра edx на 1.



7. Для вывода на экран результата вычислений используются строки: `mov eax,edx` `call iprintLF`

### #Задание для самостоятельной работы

Создаю файл `lab6-4.asm`. Проверяю создание файла.(рис. [??]).

```
(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ touch lab6-4.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ls
Makefile  image      lab6-1     lab6-1.o   lab6-2.asm lab6-3     pandoc     variant    variant.o
bib       in_out.asm lab6-1.asm lab6-2.o   lab6-3.asm lab6-4.asm report.md  variant.asm
```

### Создание файла

Открываю созданную файл и начинаю печатать в него текст программы для вычисления  $(x^3) * (1/3) + 21$  (вариант 20)(рис. [??]).

```
File Edit Search View Document Help
~ /work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab06/report/lab6-4.asm - Mousepad

1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 rem: DB 'Ответ: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintf
12 mov ecx, x
13 mov edx, 80
14 call read
15 mov eax, x
16 call atoi
17 mov ebx, eax
18 mul ebx
19 mov eax, ebx
20 mul eax
21 mov ebx, eax
22 mov ecx, 3
23 div ecx
24 mov eax, ebx
25 add eax, 21
26 mov edi, eax
27 mov eax, rem
28 call sprintf
29 mov eax, edi
30 call iprintLF
31 call quit
```

### Ввод программы

Далее сохраню файл, создаю исполняемый файл и запускаю его. (рис. [??]).

```
(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ nasm -f elf lab6-4.asm

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ld -m elf_i386 -o lab6-4 lab6-4.o

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ./lab6-4
Введите x:
1
Ответ: 22

(nvnovikov1@nvnovikov1)-[~/../arch-pc/labs/lab06/report]
$ ./lab6-4
Введите x:
3
Ответ: 30
```

### Проверка для значения $x_1$ и $x_2$



Всё верно работает.

## **5 Выводы**

После выполнения данной работы я освоил арифметические инструкции языка ассемблера NASM

## **Список литературы**