

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Архитектура компьютера

Студент: Татур Платон Андреевич

Группа: НКАбд-05-25

МОСКВА

2025 г.

6.1. Цель работы

Освоение арифметических инструкций языка ассемблера NASM

6.3. Порядок выполнения лабораторной работы

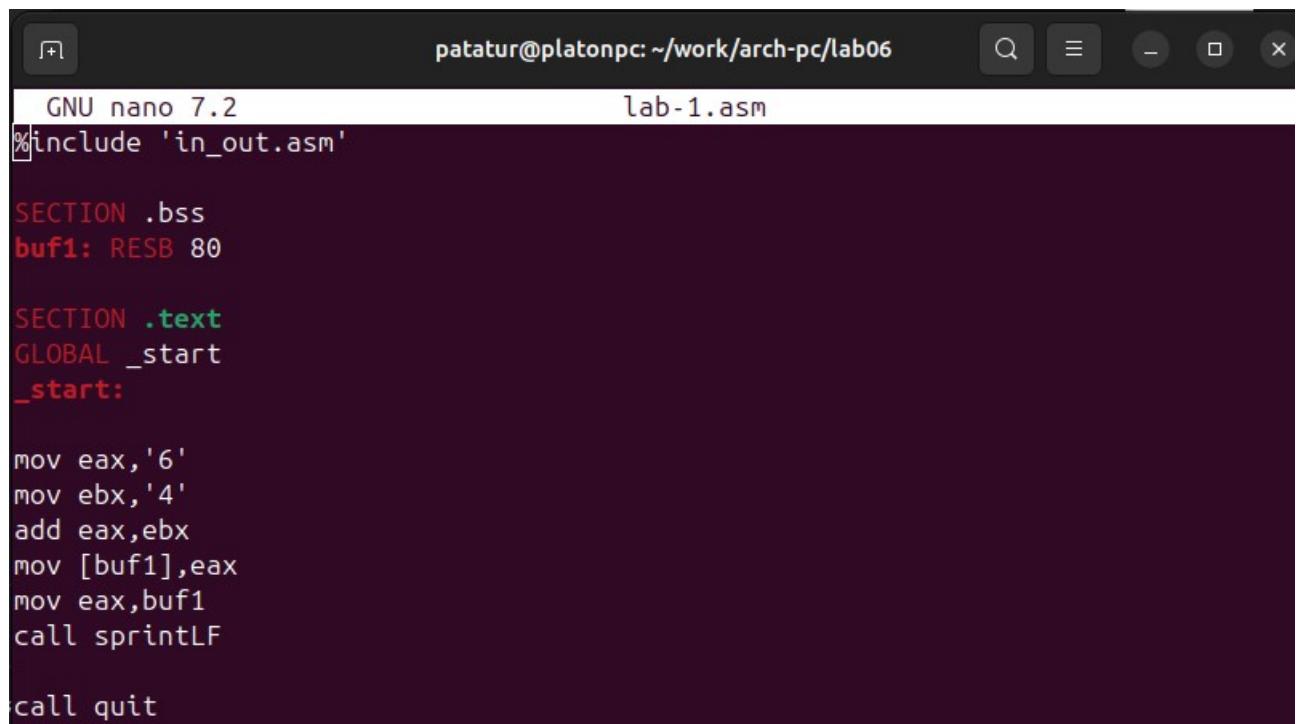
6.3.1. Символьные и численные данные в NASM

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

Рис 1

- 1) Я создал каталог для программ лабораторной работы № 6, перешел в него и создал файл lab6-1.asm (см рис 1)

Листинг 6.1. Программа вывода значения регистра eax



```
GNU nano 7.2                               lab-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

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

Рис 3

- 2) Ввел в файл lab6-1.asm текст программы из листинга 6.1 (см Рис 2)

Создал исполняемый файл и запустил его. (см Рис 3)

The terminal window title is "patatur@platonpc: ~/work/arch-pc/lab06". The file being edited is "lab6-1.asm". The assembly code is:

```
GNU nano 7.2                                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 4

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

```
patatur@platonpc:~/work/arch-pc/lab06$ 
```

Рис 5

3) Изменил текст программы lab6-1.asm и вместо символов, записал в регистры числа.

(см Рис 4)

Создал исполняемый файл и запустил его (см Рис 5)

Ответ на вопрос: Код 10 соответствует символу “LF, \n”, данный символ отвечает за перенос строки и не отображается при выводе на экран.

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

Рис 6

The terminal window title is "patatur@platonpc: ~/work/arch-pc/lab06". The file being edited is "lab6-2.asm". The assembly code is:

```
GNU nano 7.2                                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 7

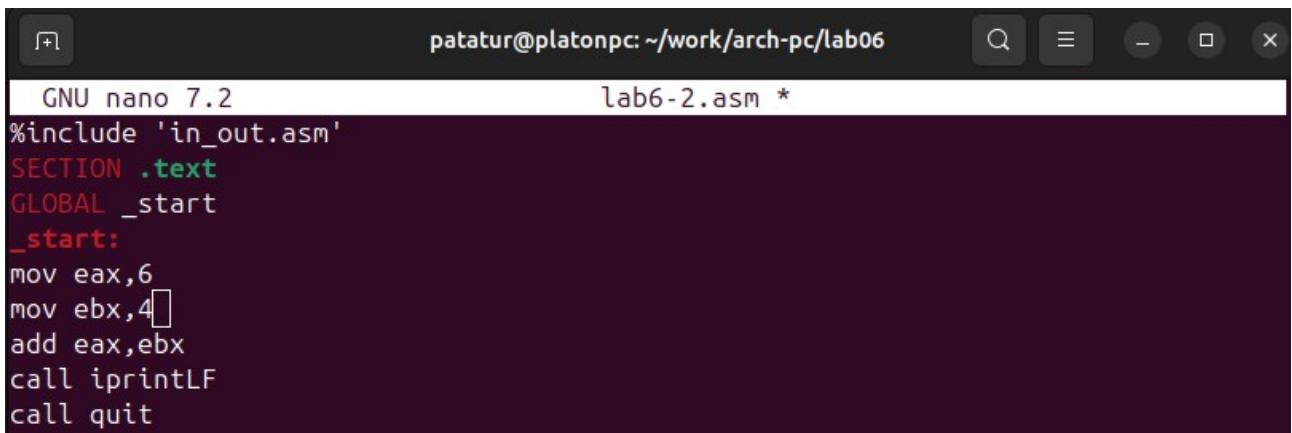
```
patatur@platonpc:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
patatur@platonpc:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
patatur@platonpc:~/work/arch-pc/lab06$ ./lab6-2
106
```

Рис 8

4) Создал файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввел в него текст программы из листинга 6.2. (см Рис 5.1)

Преобразовал текст программы из Листинга 6.1 с использованием функций из файла in_out.asm. (см Рис 5.2)

Создал исполняемый файл и запустил его. (см Рис 5.3)



The screenshot shows a terminal window titled "patatur@platonpc: ~/work/arch-pc/lab06". The window contains the assembly code for "lab6-2.asm". The code includes directives like %include 'in_out.asm', SECTION .text, and GLOBAL _start. It also contains instructions such as mov eax,6, mov ebx,4, add eax,ebx, call iprintLF, and call quit.

```
GNU nano 7.2                               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 9

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

Рис 10

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

Рис 11 (заменил iprintLF на iprint)

5) Аналогично предыдущему примеру изменил в файле lab6-2.asm символы на числа.

(см Рис 9)

Создал исполняемый файл и запустите его. В результате получил 10 (см Рис 10)

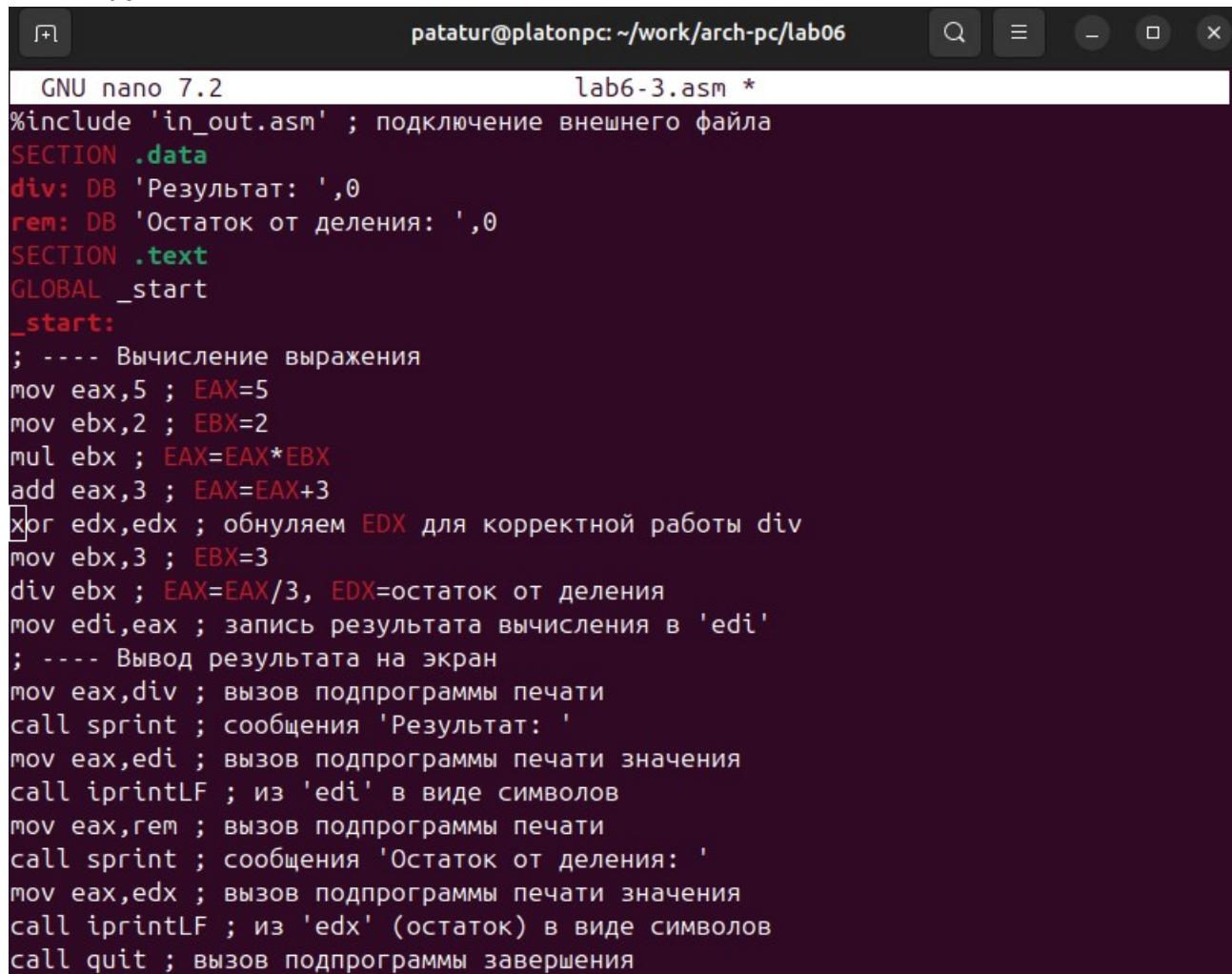
Заменил функцию iprintLF на iprint. Создал исполняемый файл и запустил его. (см Рис 10)

Вывод функций iprintLF и iprint отличается тем, что iprintLF выведет строку с переносом на следующую строку, а iprint без. (см Рис 10 и Рис 11)

6.3.2. Выполнение арифметических операций в NASM

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

Рис 12



The screenshot shows a terminal window titled "patatur@platonpc: ~/work/arch-pc/lab06". The window contains the assembly code for "lab6-3.asm". The code includes directives like %include, SECTION, GLOBAL, and _start, along with various assembly instructions such as mov, add, and call. The assembly code is written in Intel syntax.

```
GNU nano 7.2                               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
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 13

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

Рис 14

The screenshot shows a terminal window with the following content:

```
GNU nano 7.2                               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
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 15

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

Рис 16

6) Создай файл lab6-3.asm в каталоге ~/work/arch-pc/lab06. (см Рис 12)

Ввел в файл lab6-3.asm текст программы из листинга 6.3 (см Рис 13)

Создал исполняемый файл и проверил его работу. (см Рис 14)

Изменил текст программы для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (см Рис 15)

Создал исполняемый файл и проверил его работу. (см Рис 16)

Screenshot of a terminal window showing assembly code in the nano editor. The file is named 'variant.asm'. The code prompts for a student ID and calculates a variant number based on it.

```
GNU nano 7.2                               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 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,20
    div ebx
    inc edx
    mov eax,rem
    call sprint
    mov eax,edx
    call iprintLF
    call quit
```

Рис 17

Screenshot of a terminal window showing the compilation and execution of the assembly program. The user runs 'nasm -f elf variant.asm', 'ld -m elf_i386 -o variant variant.o', and then './variant'. The program prompts for a student ID and prints the calculated variant number.

```
patatur@platonpc:~/work/arch-pc/lab06$ nasm -f elf variant.asm
patatur@platonpc:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
patatur@platonpc:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032253508
Ваш вариант: 9
```

Рис 18

7) Создал файл variant.asm в каталоге ~/work/arch-pc/lab06 и ввел в него текст из листинга 6.4. (см Рис 17)

Создал исполняемый файл и запустил его (см Рис 18)

Программа выполнила вычисления правильно.

Ответы на вопросы:

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

За вывод на экран сообщения ‘Ваш вариант:’ отвечают строки:

mov eax,rem

call sprint

2. Для чего используется следующие инструкции?

mov ecx, x

mov edx, 80

call sread

Для считывания строки

3. Для чего используется инструкция “call atoi”?

Для преобразования ascii-код символа в целое число

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

За вычисление варианта отвечают строки:

mov ebx,20

div ebx

inc edx

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

В регистр eax

6. Для чего используется инструкция “inc edx”?

Данная инструкция увеличивает значение регистра edx на 1

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

За вывод на экран результатов вычислений отвечают строки:

mov eax,edx

call iprintLF

6.4. Задание для самостоятельной работы

```
patatur@platonpc:~/work/arch-pc/lab06$ cd ~/work/arch-pc/lab06
patatur@platonpc:~/work/arch-pc/lab06$ touch var9.asm
patatur@platonpc:~/work/arch-pc/lab06$ gedit var9.asm
```

Рис 19

The screenshot shows a terminal window with the following content:

```
Открыть ⌄ *var9.asm
/home/patatur/work/arch-pc/lab06 Сохранить ⌂ ⌓ ⌚ ⌚
1 %include 'in_out.asm'
2 SECTION .data
3 msg: DB 'Введите x: ',0
4 rem: DB 'Результат: ',0
5 SECTION .bss
6 x: RESB 80
7 SECTION .text
8 GLOBAL _start
9 _start:
10 mov eax, msg
11 call sprintLF
12 mov ecx, x
13 mov edx, 80
14 call sread
15 mov eax,x ; вызов подпрограммы преобразования
16 call atoi ; ASCII кода в число, `eax=x
17 xor edx,edx
18 mov ebx,31
19 mul ebx
20 sub eax,5
21 add eax,10
22 mov edi,eax
23 mov eax,rem
24 call sprint
25 mov eax,edi
26 call iprintLF
27 call quit
```

Рис 20

```
patatur@platonpc:~/work/arch-pc/lab06$ nasm -f elf var9.asm
patatur@platonpc:~/work/arch-pc/lab06$ ld -m elf_i386 -o var9 var9.o
patatur@platonpc:~/work/arch-pc/lab06$ ./var9
Введите x:
3
Результат: 98
patatur@platonpc:~/work/arch-pc/lab06$ ./var9
Введите x:
1
Результат: 36
```

Рис 21

Создал файл var9 для написания программы вычисления выражения номер 9 ($10 + (31x - 5)$), который получил при выполнения программы variant (см Рис 18) (см Рис 19)

Написал текст программы (см Рис 20)

Создал исполняемый файл и проверил его работу (см Рис 21)

Программа выполняет вычисления правильно.

Вывод

В ходе выполнения лабораторной работы я освоенил арифметические инструкций языка ассемблера NASM