

Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Бизев Никита Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Символьные и численные данные в NASM	9
4.2	Выполнение арифметических операций в NASM	16
4.2.1	Ответы на вопросы по программе	22
5	Выводы	24
	Список литературы	25

Список иллюстраций

4.1	Создание lab6-1.asm	9
4.2	Ввод данных из листинга в lab6-1.asm	10
4.3	Создание исполняемого файла	10
4.4	Запуск файла	11
4.5	Изменяю файл lab6-1-1.asm	11
4.6	Запуск файла lab6-1.asm	12
4.7	Создание lab6-2.asm	13
4.8	Ввод текста программы в файл	13
4.9	Вывод работы программы	14
4.10	Замена символов на числа	14
4.11	Вывод работы программы	15
4.12	Замена функции inprintLF на iprint	15
4.13	Запуск программы	16
4.14	Создание lab6-3.asm	17
4.15	Редактирование файла	17
4.16	Создание и запуск lab6-3	18
4.17	Редактирование файла	19
4.18	Создание и запуск lab6-3	19
4.19	Создание variant.asm	20
4.20	Ввод текста программы	21
4.21	Создание и запуск variant	22

Список таблиц

1 Цель работы

Цель данной лабораторной работы - освоение арифметических инструкций языка ассемблера NASM.

2 Задание

1. Символьные и численные данные в NASM
2. Выполнение арифметических операций в NASM
3. Выполнение заданий для самостоятельной работы

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.

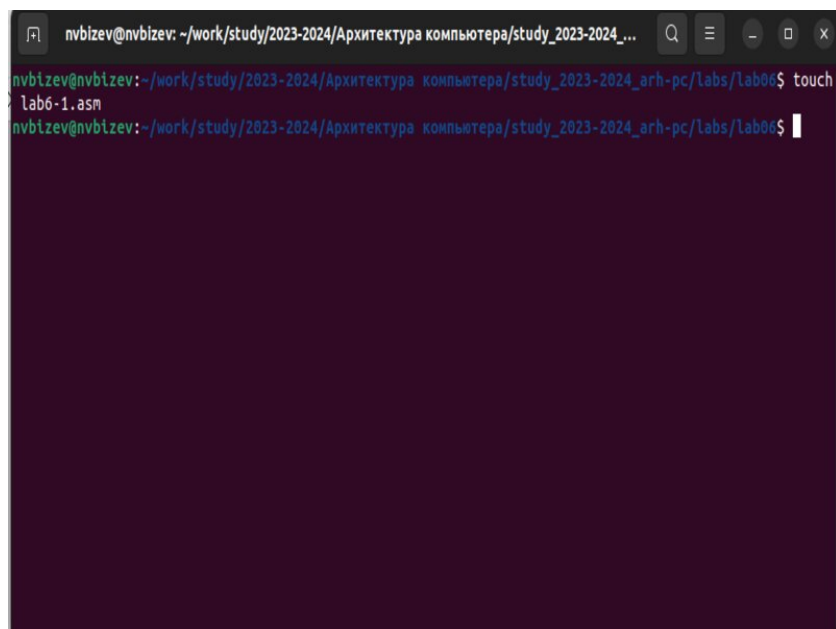
Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно 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

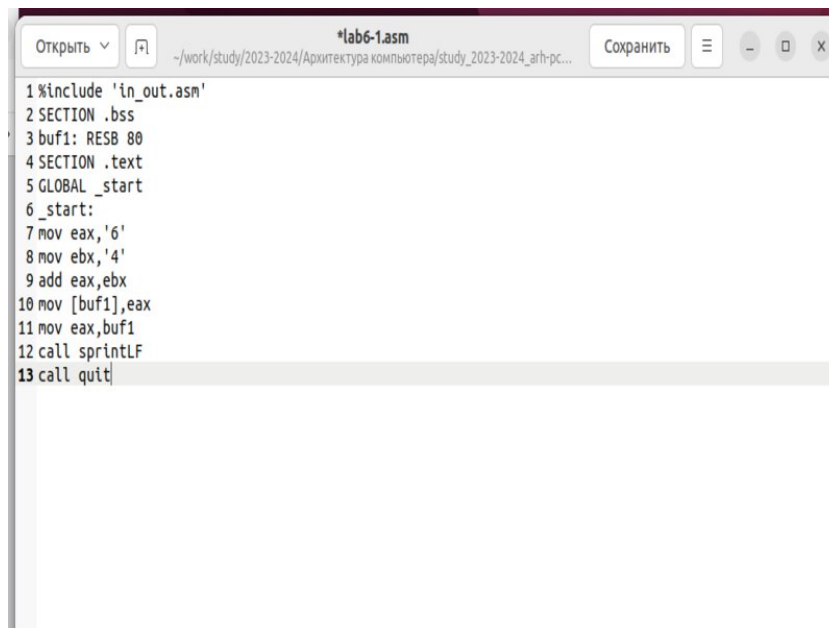
Создаю файл lab6-1.asm (рис. 4.1).

A screenshot of a terminal window with a dark background. The window title is "nvbizev@nvbizev: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_...". The terminal shows the command "touch lab6-1.asm" being entered and executed. The prompt "nvbizev@nvbizev: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06\$" is visible at the end of the line.

```
nvbizev@nvbizev: ~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_...  
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$ touch  
lab6-1.asm  
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
```

Рис. 4.1: Создание lab6-1.asm

Ввожу в файл lab6-1.asm текст программы из листинга 6.1 (рис. 4.2).

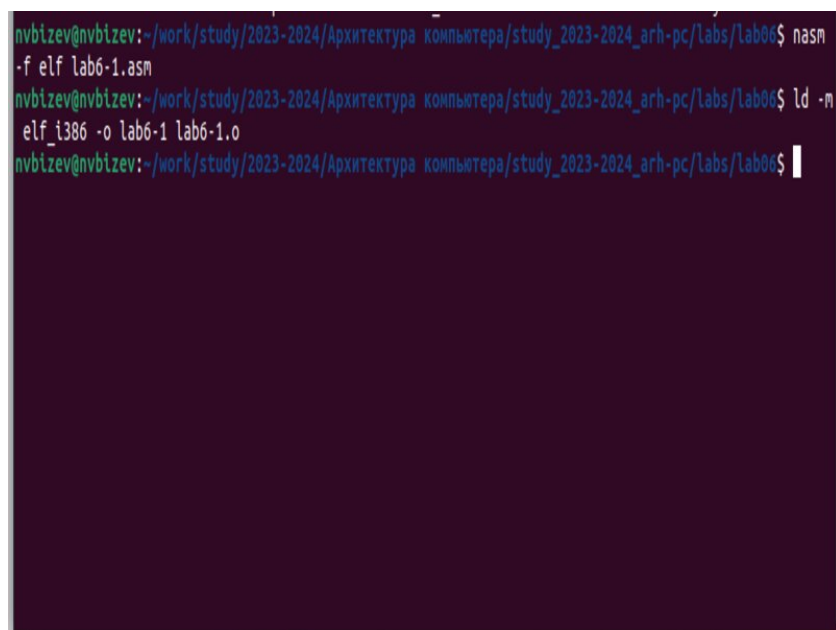


```
*lab6-1.asm
~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc...
Сохранить

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
```

Рис. 4.2: Ввод данных из листинга в lab6-1.asm

Создаю исполняемый файл lab6-1.asm (рис. 4.3).



```
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$ nasm
-f elf lab6-1.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$ ld -m
elf_i386 -o lab6-1 lab6-1.o
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
```

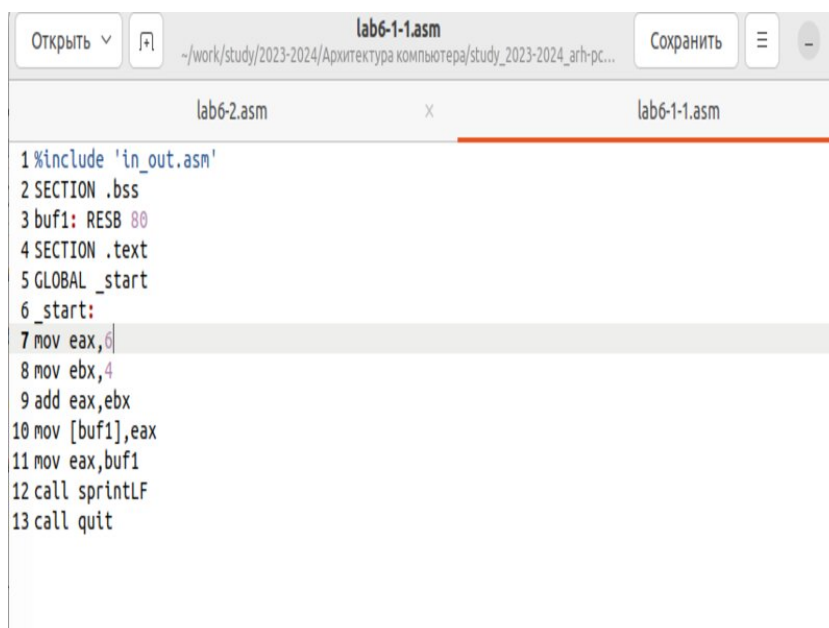
Рис. 4.3: Создание исполняемого файла

Запускаю файл. (рис. 4.4).

```
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$  
./lab6-1  
j  
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
```

Рис. 4.4: Запуск файла

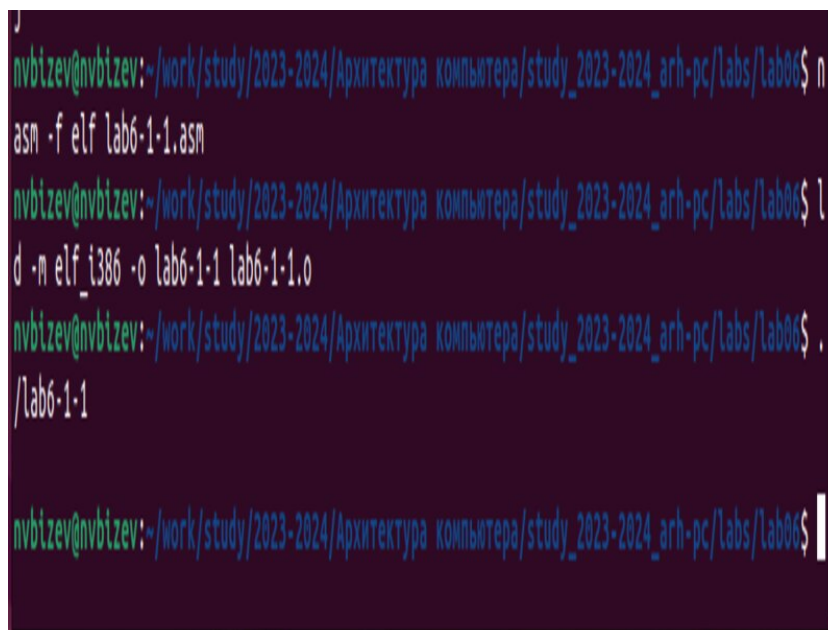
Создаю файл lab6-1-1.asm и изменяю в нем символы '6', '4' на числа 6 и 4. (рис. 4.5).



```
lab6-1-1.asm  
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
```

Рис. 4.5: Изменяю файл lab6-1-1.asm

Запускаю файл. (рис. 4.6).

A terminal window with a dark background and light green text. The prompt is 'nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06\$'. The first command is 'nasm -f elf lab6-1-1.asm'. The second command is 'ld -n elf_i386 -o lab6-1-1 lab6-1-1.o'. The third command is './lab6-1-1'. The prompt is shown again at the bottom.

```
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$ nasm -f elf lab6-1-1.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$ ld -n elf_i386 -o lab6-1-1 lab6-1-1.o
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$ ./lab6-1-1
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
```

Рис. 4.6: Запуск файла lab6-1.asm

Теперь вывелся символ с кодом 10, это символ перевода строки, этот символ не отображается при выводе на экран.

Создаю файл lab6-2.asm (рис. 4.7).

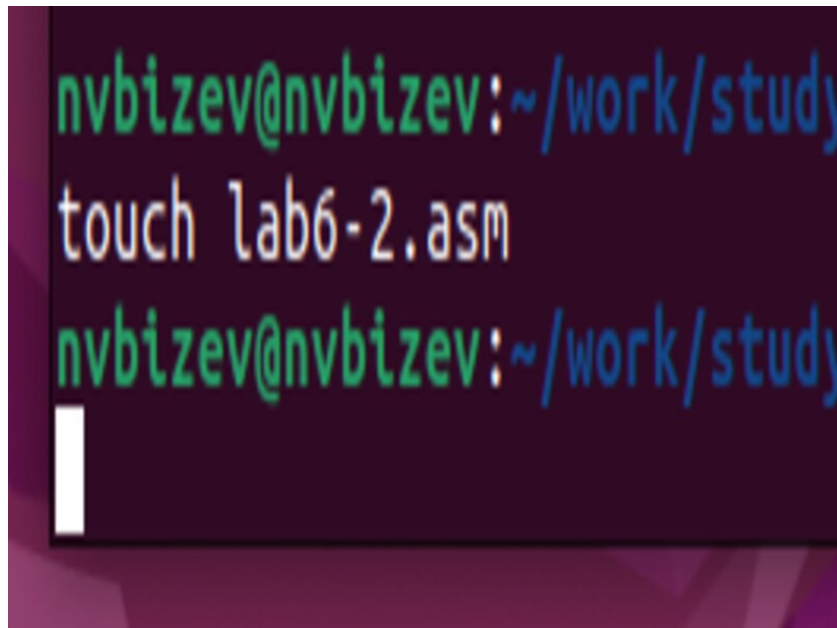


Рис. 4.7: Создание lab6-2.asm

Ввожу в файл lab6-2.asm текст программы из листинга 6.2.(рис. 4.8)

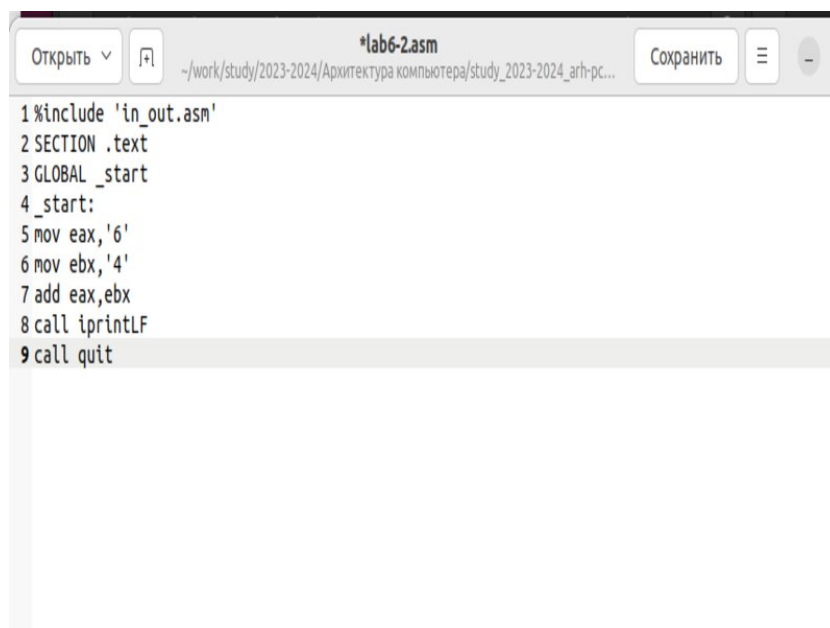



Рис. 4.8: Ввод текста программы в файл

Создаю исполняемый файл и вывожу результат работы программы. (рис. 4.9).

```
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-  
nasm -f elf lab6-2.asm  
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-  
ld -m elf_i386 -o lab6-2 lab6-2.o  
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-  
./lab6-2  
106  
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-
```

Рис. 4.9: Вывод работы программы

Аналогично предыдущему примеру изменяю символы на числа. (рис. 4.10).



The screenshot shows a text editor window titled '*lab6-2.asm'. The menu bar includes 'Открыть' (Open), a file icon, the filename, the path '~/.work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-пс...', and 'Сохранить' (Save). The code content is as follows:

```
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
```

Рис. 4.10: Замена символов на числа

Вывожу результат с заменой. (рис. 4.11).

```

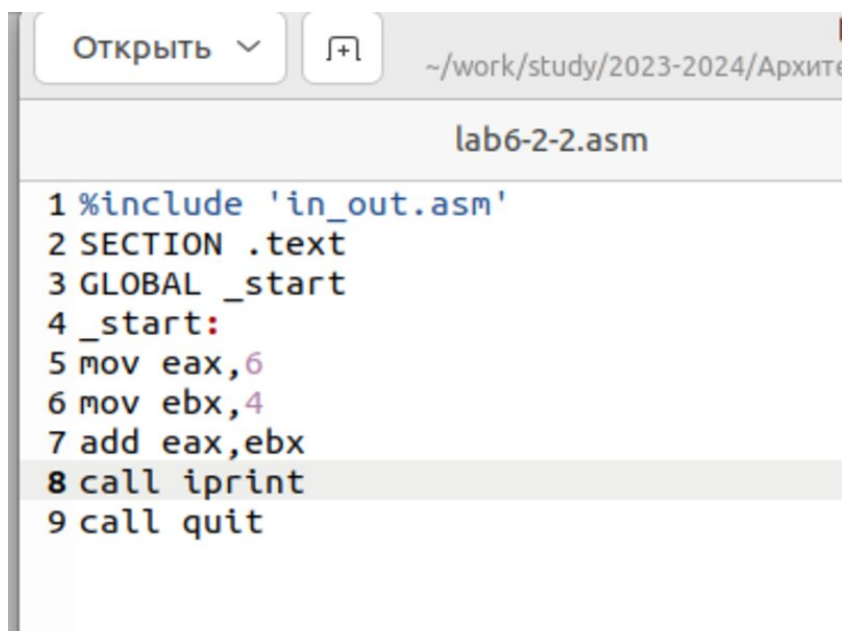
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
nasm -f elf lab6-2-2.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
ld -m elf_i386 -o lab6-2-2 lab6-2-2.o
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$
./lab6-2-2
10
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab06$

```

Рис. 4.11: Вывод работы программы

Результат работы программы - 10.

Создаю файл lab6-2-2 и заменяю функцию `inprintLF` на `iprint`. (рис. 4.12).



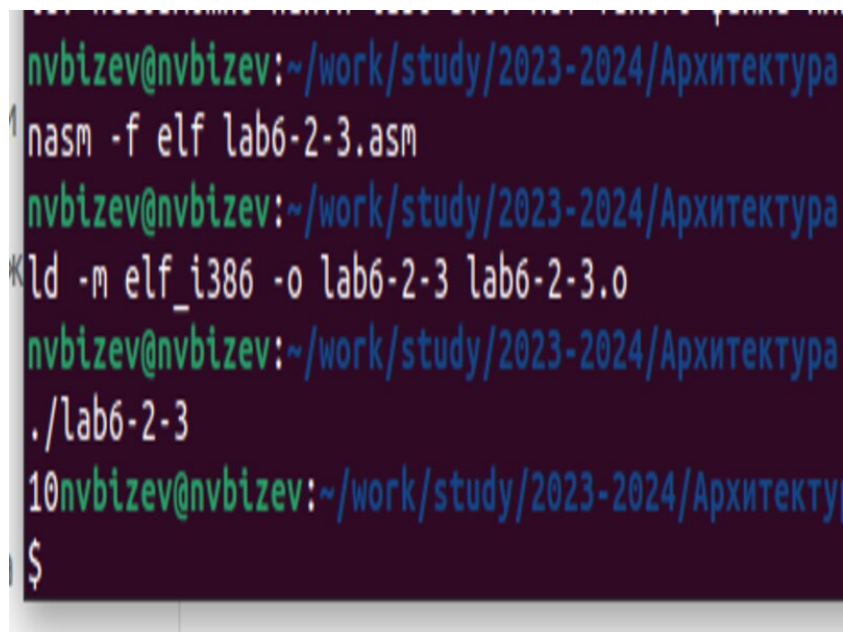
```

Открыть ▾  ~/work/study/2023-2024/Архитектура компьютера/labs/lab06/lab6-2-2.asm
lab6-2-2.asm
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

```

Рис. 4.12: Замена функции `inprintLF` на `iprint`

Запускаю программу. (рис. 4.13).



```
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура
1 nasm -f elf lab6-2-3.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура
2 ld -m elf_i386 -o lab6-2-3 lab6-2-3.o
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура
3 ./lab6-2-3
10 nvbizev@nvbizev:~/work/study/2023-2024/Архитекту
$
```

Рис. 4.13: Запуск программы

При использовании функции `iprintLF` результат выводится на следующую строку, в то время как при использовании функции `iprint` результат выводится сразу.

4.2 Выполнение арифметических операций в NASM

Создаю файл `lab6-3.asm` (рис. 4.14).

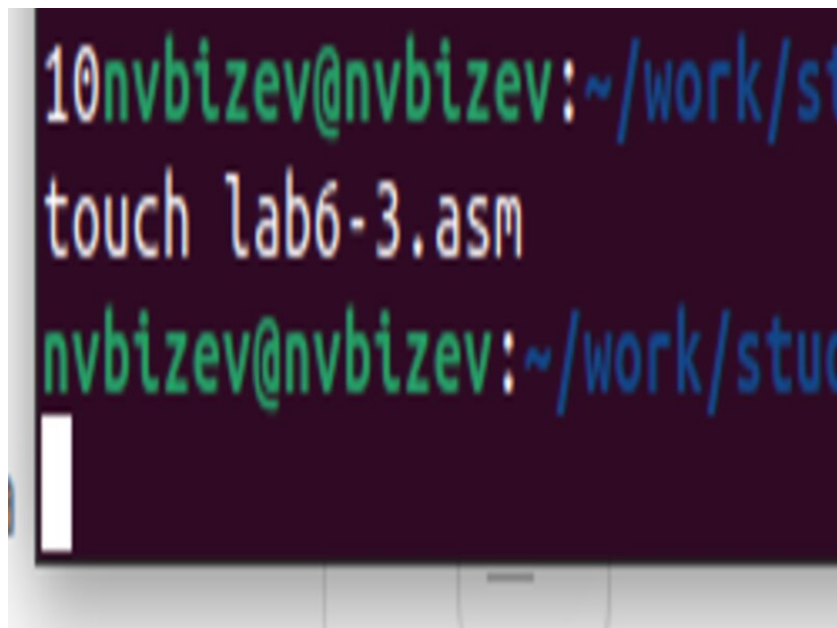


Рис. 4.14: Создание lab6-3.asm

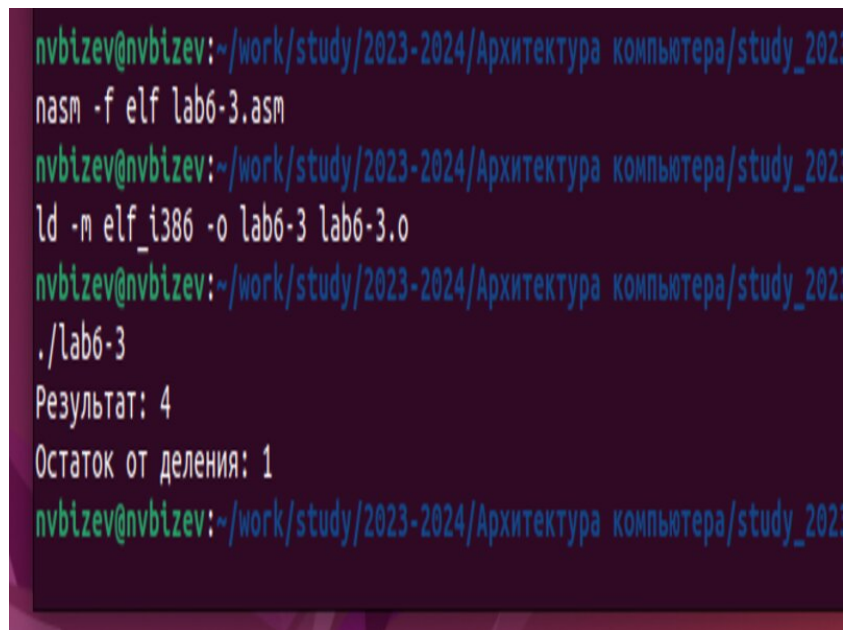
Открываю файл в текстовом редакторе и ввожу текст программы из листинга 6.3.(рис. 4.15).

A screenshot of a text editor window titled 'lab6-3.asm'. The window shows assembly code with line numbers 1 through 26. The code includes comments in Russian and assembly instructions. The editor has a menu bar with 'Открыть' and 'Сохранить', and a toolbar with icons for file operations. The code is as follows:

```
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 iprintf ; 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 iprintf ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения
```

Рис. 4.15: Редактирование файла

Создаю исполняемый файл lab6-3 и запускаю его. (рис. 4.16).

A terminal window with a dark background and light-colored text. The prompt is 'nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024/'. The user enters 'nasm -f elf lab6-3.asm'. The prompt repeats. The user enters 'ld -m elf_i386 -o lab6-3 lab6-3.o'. The prompt repeats. The user enters './lab6-3'. The prompt repeats. The output shows 'Результат: 4' and 'Остаток от деления: 1'. The prompt repeats.

```
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024/
nasm -f elf lab6-3.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024/
ld -m elf_i386 -o lab6-3 lab6-3.o
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024/
./lab6-3
Результат: 4
Остаток от деления: 1
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024/
```

Рис. 4.16: Создание и запуск lab6-3

Результат программы удовлетворяет $f(x) = (5 * 2 + 3)/3$. Теперь я заменяю это выражение на $f(x) = (4 * 6 + 2)/5$.

Изменяю файл чтобы он удовлетворял выше указанному выражению. (рис. 4.17).

```

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,4 ; EAX=4
10 mov ebx,6 ; EBX=6
11 mul ebx ; EAX=EAX*EBX
12 add eax,2 ; EAX=EAX+2
13 xor edx,edx ; обнуляем EDX для корректной работы div
14 mov ebx,5 ; EBX=5
15 div ebx ; EAX=EAX/5, 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 iprintf ; 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 iprintf ; из 'edx' (остаток) в виде символов
26 call quit ; вызов подпрограммы завершения

```

Рис. 4.17: Редактирование файла

Создаю исполняемый файл lab6-3 и запускаю его. (рис. 4.18).

```

nvbizhev@nvbizhev:~/work/study/2023-2024/Архитектура комп
nasm -f elf lab6-3-3.asm
nvbizhev@nvbizhev:~/work/study/2023-2024/Архитектура комп
ld -m elf_i386 -o lab6-3-3 lab6-3-3.o
nvbizhev@nvbizhev:~/work/study/2023-2024/Архитектура комп
./lab6-3-3
Результат: 5
Остаток от деления: 1
nvbizhev@nvbizhev:~/work/study/2023-2024/Архитектура комп

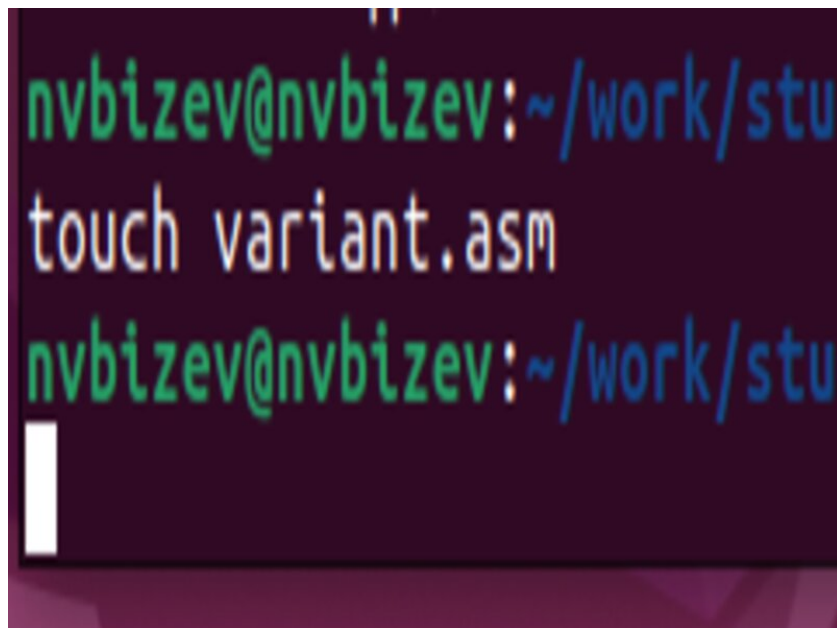
```

Рис. 4.18: Создание и запуск lab6-3

Результат работы - 5, остаток же - 1.

Рассмотрим другой пример.

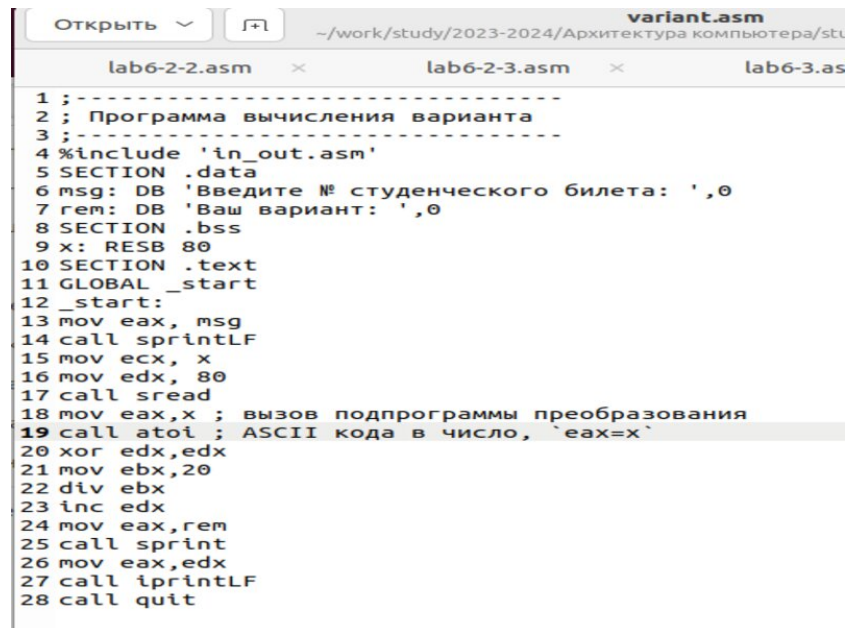
Создаю файл variant.asm (рис. 4.19).

A terminal window with a dark purple background. The prompt 'nvbizev@nvbizev:~/work/stu' is shown in green. The command 'touch variant.asm' is entered in white. The prompt 'nvbizev@nvbizev:~/work/stu' is shown again in green, followed by a white cursor. The terminal has a thin black border.

```
nvbizev@nvbizev:~/work/stu
touch variant.asm
nvbizev@nvbizev:~/work/stu
█
```

Рис. 4.19: Создание variant.asm

Ввожу текст программы из листинга 6.4 в файл. (рис. 4.20).



```
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 sprintLF  
15 mov ecx, x  
16 mov edx, 80  
17 call sread  
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 iprintLF  
28 call quit
```

Рис. 4.20: Ввод текста программы

Программа запрашивает № студенческого билета и основываясь на нем выводит мой вариант. Номер моего студенческого билет - 1132230806.

Создаю исполняемый файл variant и запускаю его. (рис. 4.21).

```
code: variant.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура/variant
nasm -f elf variant.asm
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура/variant
ld -m elf_i386 -o variant variant.o
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура/variant
./variant
Введите № студенческого билета:
1132230806
Ваш вариант: 7
nvbizev@nvbizev:~/work/study/2023-2024/Архитектура/variant
```

Рис. 4.21: Создание и запуск variant

4.2.1 Ответы на вопросы по программе

1. За вывод сообщения “Ваш вариант” отвечают строки кода:

```
mov eax, rem
call sprint
```

2. Инструкция `mov ecx, x` используется, чтобы положить адрес вводимой строки `x` в регистр `ecx` `mov edx, 80` - запись в регистр `edx` длины вводимой строки `call sread` - вызов подпрограммы из внешнего файла, обеспечивающей ввод сообщения с клавиатуры
3. `call atoi` используется для вызова подпрограммы из внешнего файла, которая преобразует `ascii`-код символа в целое число и записывает результат в регистр `eax`
4. За вычисления варианта отвечают строки:

```
xor edx,edx ; обнуление edx для корректной работы div
mov ebx,20 ; ebx = 20
div ebx ; eax = eax/20, edx - остаток от деления
inc edx ; edx = edx + 1
```

5. При выполнении инструкции `div ebx` остаток от деления записывается в регистр `edx`
6. Инструкция `inc edx` увеличивает значение регистра `edx` на 1
7. За вывод на экран результатов вычислений отвечают строки:

```
mov eax,edx
call iprintLF
```

5 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM.

Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2089086/mod_resource/content/0/%D0%9B%D0%B0