

Отчёт по лабораторной работе №7

дисциплина: Архитектура компьютера

Максим Александрович Мишонков

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	7
5	Выводы	18

Список иллюстраций

4.1	Создание каталога и файла	7
4.2	Введение текста программы	7
4.3	Проверка работы исполняемого файла	7
4.4	Изменение текста программы	8
4.5	Проверка работы исполняемого файла	8
4.6	Создание файла	8
4.7	Введение текста программы	9
4.8	Проверка работы исполняемого файла	9
4.9	Изменение текста программы	10
4.10	Проверка работы исполняемого файла	10
4.11	Изменение текста программы	11
4.12	Проверка работы исполняемого файла	11
4.13	Создание файла	11
4.14	Введение текста программы	12
4.15	Проверка работы исполняемого файла	12
4.16	Изменение текста программы	12
4.17	Проверка работы исполняемого файла	13
4.18	Создание файла	13
4.19	Введение текста программы	13
4.20	Проверка работы исполняемого файла	14
4.21	Создание файла	15
4.22	Написание текста программы	16
4.23	Написание текста программы	16
4.24	Проверка работы программы	17

1 Цель работы

Целью данной лабораторной работы является освоение арифметических инструкций языка ассемблера NASM.

2 Задание

Освоить арифметические инструкции языка ассемблера 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.

Существует три основных способа адресации:

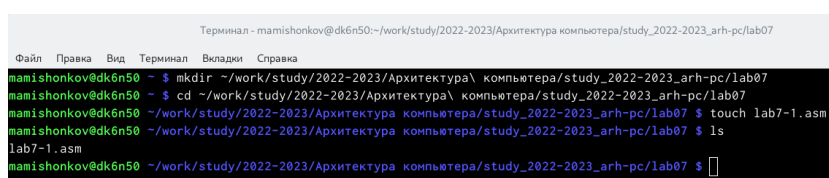
Регистровая адресация – операнды хранятся в регистрах, и в команде используются имена этих регистров. Например: `mov ax,bx`.

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

4 Выполнение лабораторной работы

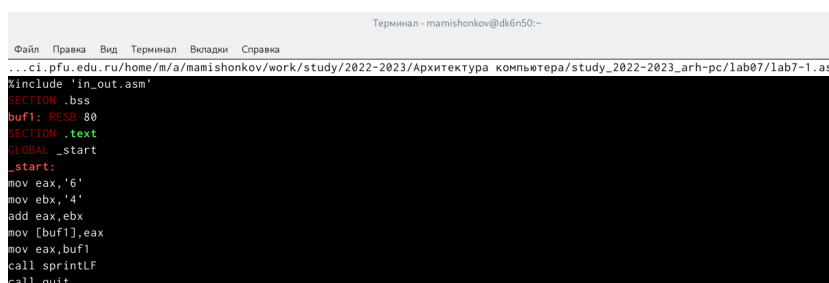
1. Создал каталог для программ лабораторной работы №7, перешёл в него и создал файл lab7-1.asm. (рис. 4.1)



```
Терминал - mamishonkov@dk6n50:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07
mamishonkov@dk6n50 ~ $ mkdir ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07
mamishonkov@dk6n50 ~ $ cd ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ touch lab7-1.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ls
lab7-1.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.1: Создание каталога и файла

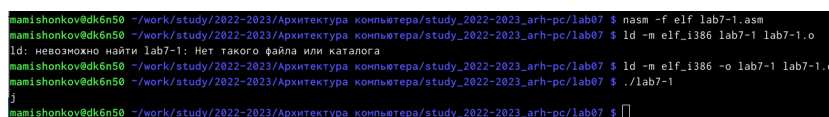
2. Ввёл в файл lab7-1.asm текст программы из листинга 7.1. (рис. 4.2)



```
Терминал - mamishonkov@dk6n50:~
mamishonkov@dk6n50 ~ $ cat ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07/lab7-1.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. 4.2: Введение текста программы

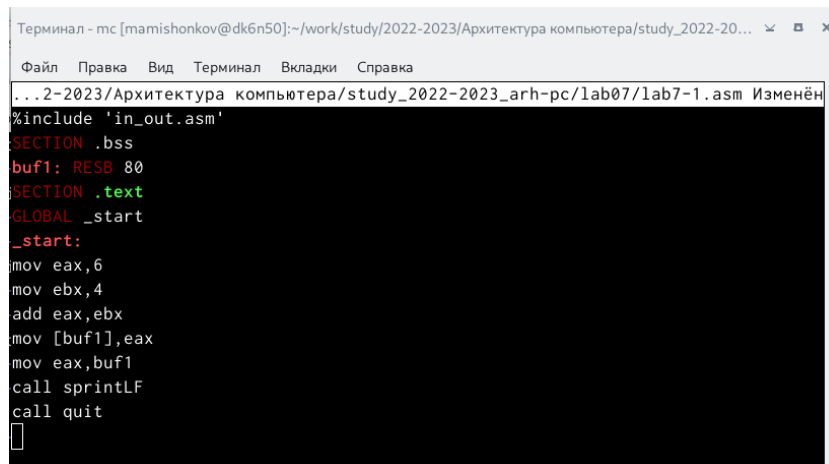
3. Создал исполняемый файл и запустил его. Программа вывела символ j. (рис. 4.3)



```
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-1.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 lab7-1 lab7-1.o
ld: невозможно найти lab7-1: Нет такого файла или каталога
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-1
j
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.3: Проверка работы исполняемого файла

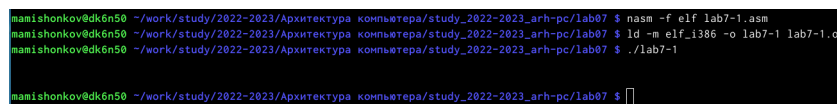
4. Изменил текст программы, вместо символов записав в регистры числа. (рис. 4.4)



```
Терминал - mc [mamishonkov@dk6n50]:~/work/study/2022-2023/Архитектура компьютера/study_2022-20...
Файл  Правка  Вид  Терминал  Вкладки  Справка
...2-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07/lab7-1.asm Изменён
#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. 4.4: Изменение текста программы

5. Создал исполняемый файл и запустил его. Программа вывела невидимый символ с кодом 10. (рис. 4.5)



```
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-1.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-1
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.5: Проверка работы исполняемого файла

6. Создал файл lab7-2.asm в каталоге lab07 и ввёл в него текст программы из листинга 7.2. (рис. 4.6, 4.7)

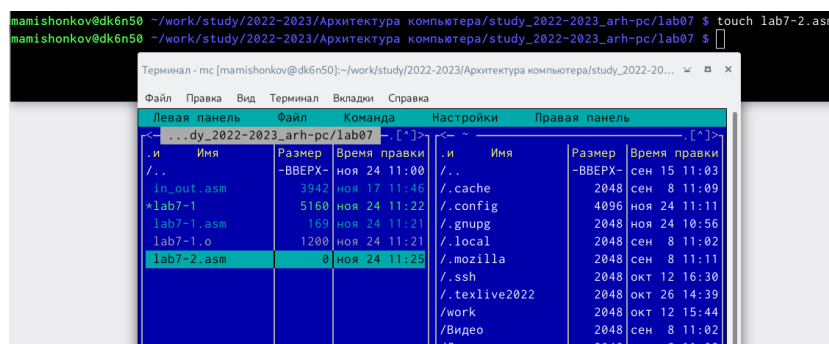
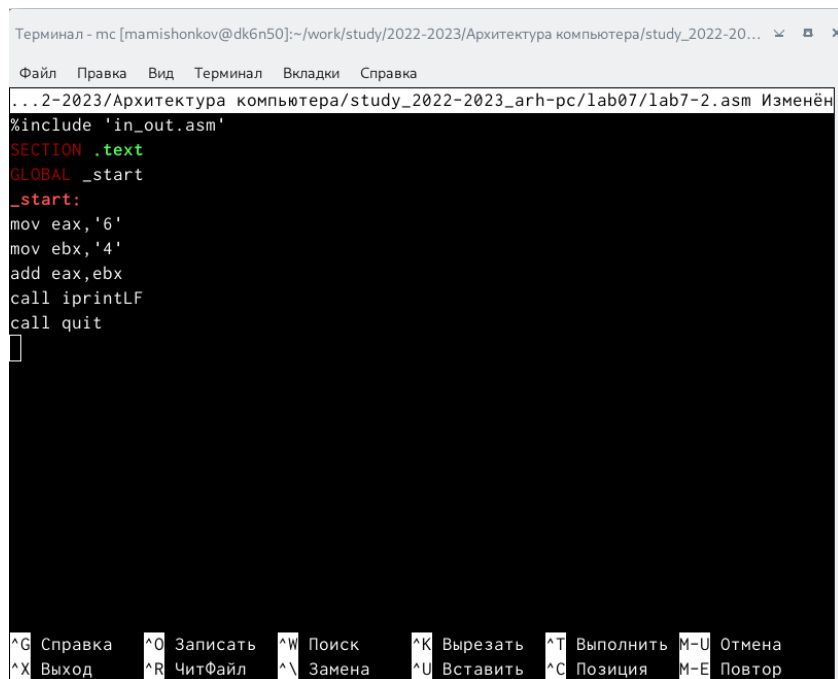


Рис. 4.6: Создание файла



```
Терминал - mc [mamishonkov@dk6n50]:~/work/study/2022-2023/Архитектура компьютера/study_2022-20...
Файл  Правка  Вид  Терминал  Вкладки  Справка
...2-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07/lab7-2.asm Изменён
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit

```

Рис. 4.7: Введение текста программы

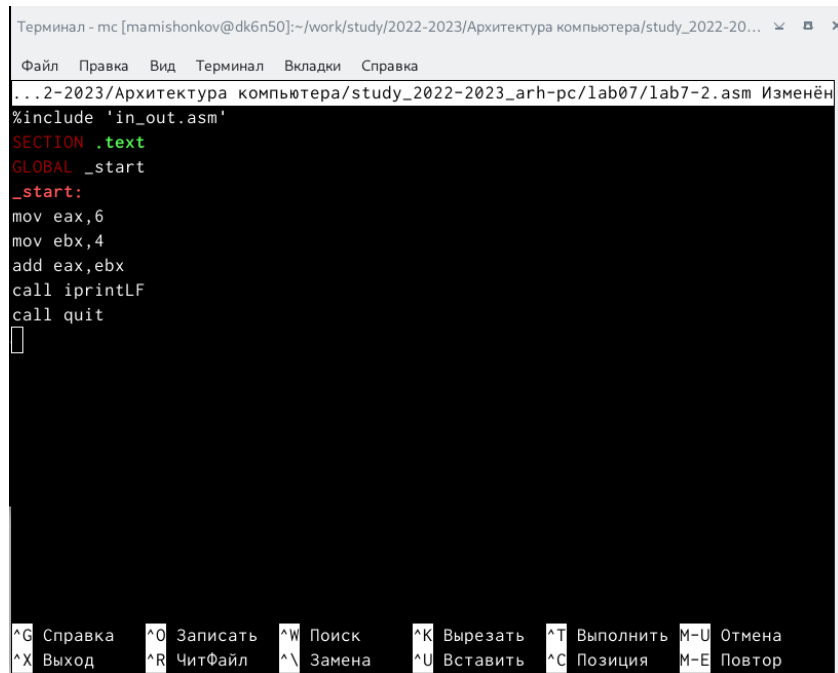
7. Создал исполняемый файл и запустил его. Программа вывела число 106.
(рис. 4.8)



```
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-2.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-2
106
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.8: Проверка работы исполняемого файла

8. Изменил текст программы, вместо символов записав в регистры числа. (рис. 4.9)

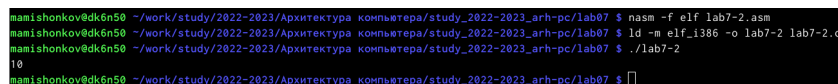


```
Терминал - mc [mamishonkov@dk6n50]:~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07/lab7-2.asm Изменён
Файл  Правка  Вид  Терминал  Вкладки  Справка
~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07/lab7-2.asm
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprintLF
call quit

```

Рис. 4.9: Изменение текста программы

9. Создал исполняемый файл и запустил его. Программа вывела число 10. (рис. 4.10)



```
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-2.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-2
10
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.10: Проверка работы исполняемого файла

10. Заменял в тексте программы функцию iprintLF на iprint. (рис. 4.11)

```

Терминал - mc [mamishonkov@dk6n50]:~/work/study/2022-2023/Архитектура компьютера/study_2022-20...
Файл  Правка  Вид  Терминал  Вкладки  Справка

...2-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07/lab7-2.asm  Изменён
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,6
mov ebx,4
add eax,ebx
call iprint
call quit

^G Справка  ^O Записать  ^W Поиск  ^K Вырезать  ^T Выполнить  M-U Отмена
^X Выход    ^R ЧитФайл  ^\ Замена  ^U Вставить  ^C Позиция  M-E Повтор

```

Рис. 4.11: Изменение текста программы

11. Создал исполняемый файл и запустил его. Результат отличается в выводе данных: при команде `iprintLF` мы начинаем вводить команду на следующей строке, а при использовании команды `iprint` мы вводим данные на той же строке. (рис. 4.12)

```

mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-2.asm
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-2
10mamishonkov@dk6n50 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $

```

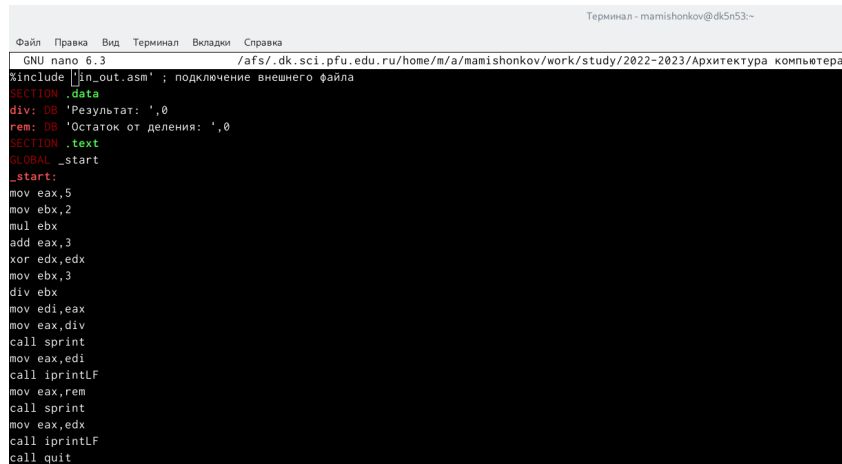
Рис. 4.12: Проверка работы исполняемого файла

12. Создал файл `lab7-3.asm` в каталоге `lab07`. (рис. 4.13)

Левая панель			Правая панель		
Имя	Размер	Время правки	Имя	Размер	Время правки
./	-ВВЕРХ-	ноя 24 11:00	./	-ВВЕРХ-	сен 15 11:03
in_out.asm	3942	ноя 17 11:46	./cache	2048	сен 8 11:09
*lab7-1	5168	ноя 24 11:22	./config	4096	ноя 24 13:28
lab7-1.asm	169	ноя 24 11:21	./gnupg	2048	ноя 24 13:45
lab7-1.o	1200	ноя 24 11:21	./local	2048	сен 8 11:02
*lab7-2	5088	ноя 24 11:43	./mozilla	2048	сен 8 11:11
lab7-2.asm	112	ноя 24 11:42	./ssh	2048	окт 12 16:38
lab7-2.o	1040	ноя 24 11:43	./texlive2022	2048	окт 26 14:39
lab7-3.asm	0	ноя 24 14:32	./work	2048	окт 12 15:44

Рис. 4.13: Создание файла

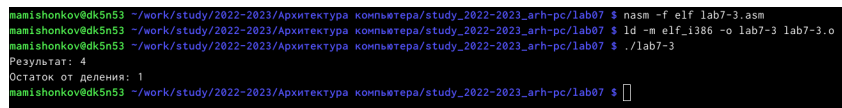
13. Ввёл в lab7-3.asm. текст программы из листинга 7.3. (рис. 4.14)



```
Терминал - mamishonkov@dk5n53:~
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov/work/study/2022-2023/Архитектура компьютера
#include "in_out.asm" ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx
mov edi,eax
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 4.14: Введение текста программы

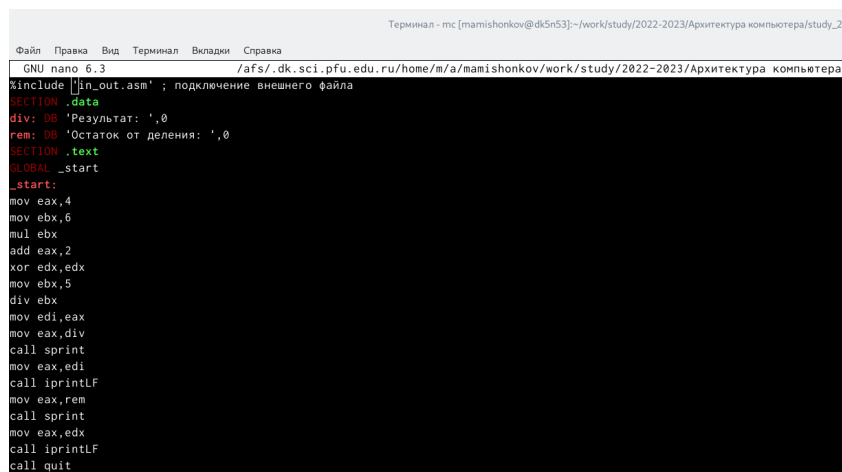
14. Создал исполняемый файл и проверил его работу. (рис. 4.15)



```
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-3.asm
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-3
Результат: 4
Остаток от деления: 1
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.15: Проверка работы исполняемого файла

15. Изменил текст программы для вычисления выражения данного выражения.
Создал исполняемый файл и проверил его работу. (рис. 4.16, 4.17)



```
Терминал - mc [mamishonkov@dk5n53]~/work/study/2022-2023/Архитектура компьютера/study_2
GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov/work/study/2022-2023/Архитектура компьютера
#include "in_out.asm" ; подключение внешнего файла
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx
mov edi,eax
mov eax,div
call sprint
mov eax,edi
call iprintLF
mov eax,rem
call sprint
mov eax,edx
call iprintLF
call quit
```

Рис. 4.16: Изменение текста программы

```

mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-3.asm
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-3
Результат: 5
Остаток от деления: 1
mamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $

```

Рис. 4.17: Проверка работы исполняемого файла

16. Создал файл variant.asm в каталоге lab07 и ввёл в него текст программы из листинга 7.4. (рис. 4.18, 4.19)

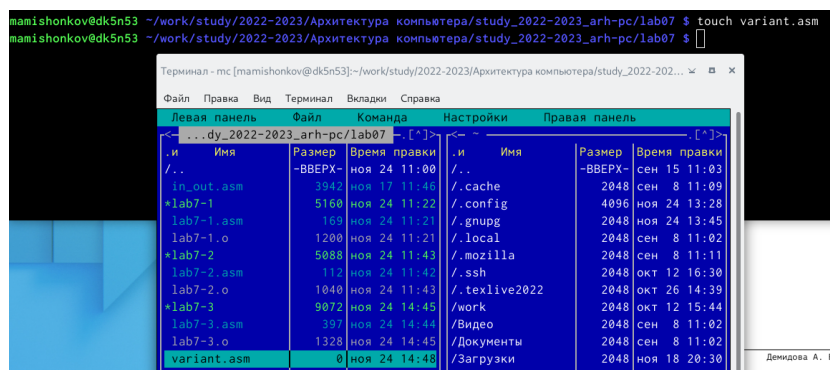


Рис. 4.18: Создание файла

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov/work/study/2022-2023/Архитектура компьютера
#include "in_out.asm"
SECTION .data
msg: DB 'Введите No 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
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx
mov eax, rem
call sprint
mov eax, edx
call iprintf
call quit

```

Рис. 4.19: Введение текста программы

17. Проверив работу исполняемого файла, я ввёл номер моего студенческого билета. Программа вывела ответ 14. Я проверил это аналитически, ответ совпал. (рис. 4.20)

```

hamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf variant.asm
hamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o variant variant.o
hamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./variant
Введите No студенческого билета:
1132226473
Ваш вариант: 14
hamishonkov@dk5n53 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $

```

Рис. 4.20: Проверка работы исполняемого файла

Вопросы

1. Какие строки из листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

Строки `mov eax, ecx` `call sprint`.

2. Для чего используются инструкции `nasm “mov ecx, x”`, `“mov edx, 80”`, `“call sread”`?

Инструкция `“mov ecx, x”` записывает адреса выводимого сообщения в ‘EAX’
Инструкция `“mov edx, 80”` записывает длину вводимого сообщения в ‘EBX’
Инструкция `“call sread”` выполняет вызов программы ввода сообщений.

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

Эта инструкция используется для преобразования кода переменной ASCII в число.

4. Какие строки из листинга 7.4 отвечают за вычисления варианта?

Строки `xor edx, edx` `mov ebx, 20` `div ebx`, `inc edx`.

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

В регистр `ebx`.

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

Эта инструкция используется для увеличения значения `edx` на единицу.

7. Какие строки из листинга 7.4 отвечают за вывод на экран результата вычислений?

Строки `mov eax, edx call iprintLF`.

Самостоятельная работа

1. Создал файл, в котором буду писать программу. (рис. 4.21)

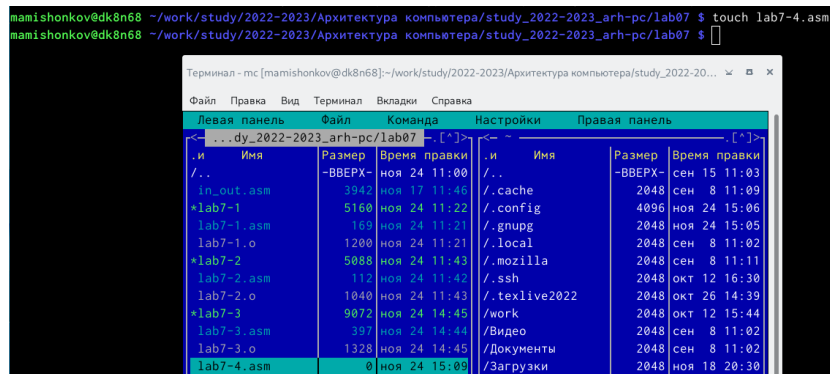


Рис. 4.21: Создание файла

2. Написал программу, которая будет решать выражение и выводить ответ при разных значениях переменной x . Так как в пункте 17 программа выдала 14 вариант, программу я писал для выражения из списка, соответствующее варианту 14. (рис. 4.22, 4.23)

```

GNU nano 6.3 /afs/.dk.sci.pfu.edu.ru/home/m/a/mamishonkov.
%include 'in_out.asm'

SECTION .data
prim: DB '(x/2+8)*3',0
x1: DB 'Введите значение x: ',0
otv1: DB 'Ответ: ',0

SECTION .bss
p: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,prim
call sprintLF

mov eax,x1
call sprint

mov ecx,p
mov edx,80
call sread

mov eax,p
call atoi

xor edx,edx

```

Рис. 4.22: Написание текста программы

```

mov ebx,2
div ebx
add eax,8
xor ebx,ebx
mov ebx,3
mul ebx

mov edi,eax

mov eax,otv1
call sprintLF
mov eax,edi
call iprintLF

call quit

```

Рис. 4.23: Написание текста программы

3. Проверил результат работы этой программы. Стоит отметить, что так как

при выполнении деления программа в качестве результата использует целую часть, то в первом случае дробь $1/2$ округляется до 0, следовательно, был получен ответ 24, а не 25. (рис. 4.24)

```
mamishonkov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ nasm -f elf lab7-4.asm
mamishonkov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
mamishonkov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-4
(x/2+8)*3
Введите значение x: 1
Ответ:
24
mamishonkov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $ ./lab7-4
(x/2+8)*3
Введите значение x: 4
Ответ:
30
mamishonkov@dk3n37 ~/work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/lab07 $
```

Рис. 4.24: Проверка работы программы

5 Выводы

В ходе выполнения данной лабораторной работы я освоил арифметические инструкции языка ассемблера NASM.