

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: Тимаков Макарий Владимирович

Группа: НПИъд 02-25

МОСКВА

2025г.

1.Цель работы Освоить арифметические инструкции языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

2. Выполнение лабораторной работы

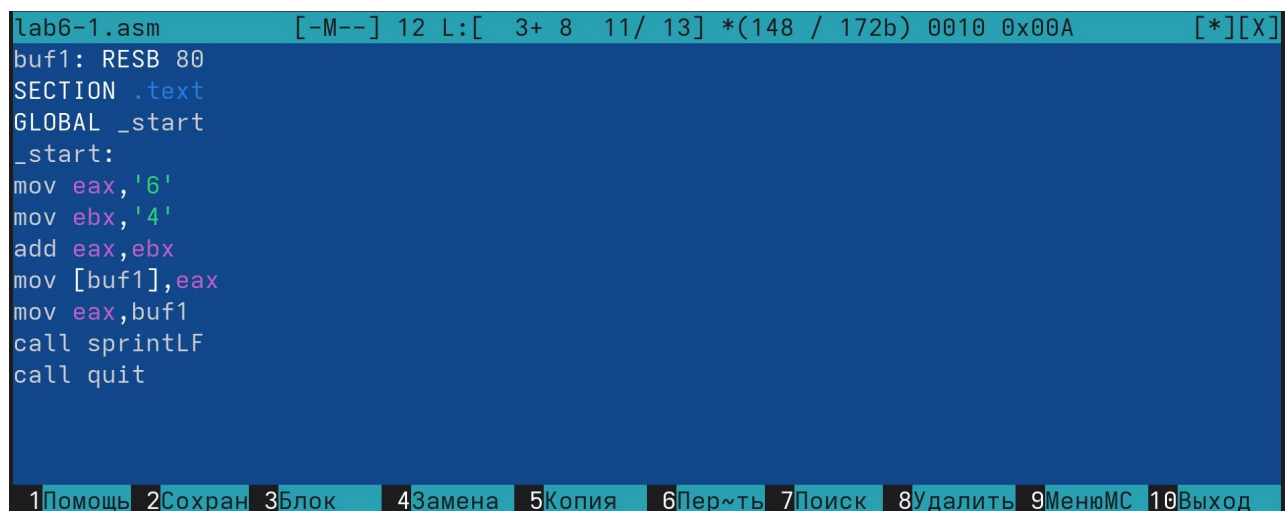
2.1 Порядок выполнения лабораторной работы

Создаем каталог для программ ЛБ6, и в нем создаем файл (Рисунок 2.1).

```
timakovmv@timakovmv:~/work/arch-pc/lab05$ mkdir ~/work/arch-pc/lab06
timakovmv@timakovmv:~/work/arch-pc/lab05$ cd ~/work/arch-pc/lab06
timakovmv@timakovmv:~/work/arch-pc/lab06$ touch lab6-1.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 2.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (Рисунок 2.2).



```
lab6-1.asm [-M--] 12 L:[ 3+ 8 11/ 13] *(148 / 172b) 0010 0x00A [*][X]
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
mov [buf1],eax
mov eax,buf1
call printf
call quit
```

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Удалить 9МенюМС 10Выход

Рисунок 2.2

Копируем файл in_out.asm в новый каталог lab06 (Рисунок 2.3)

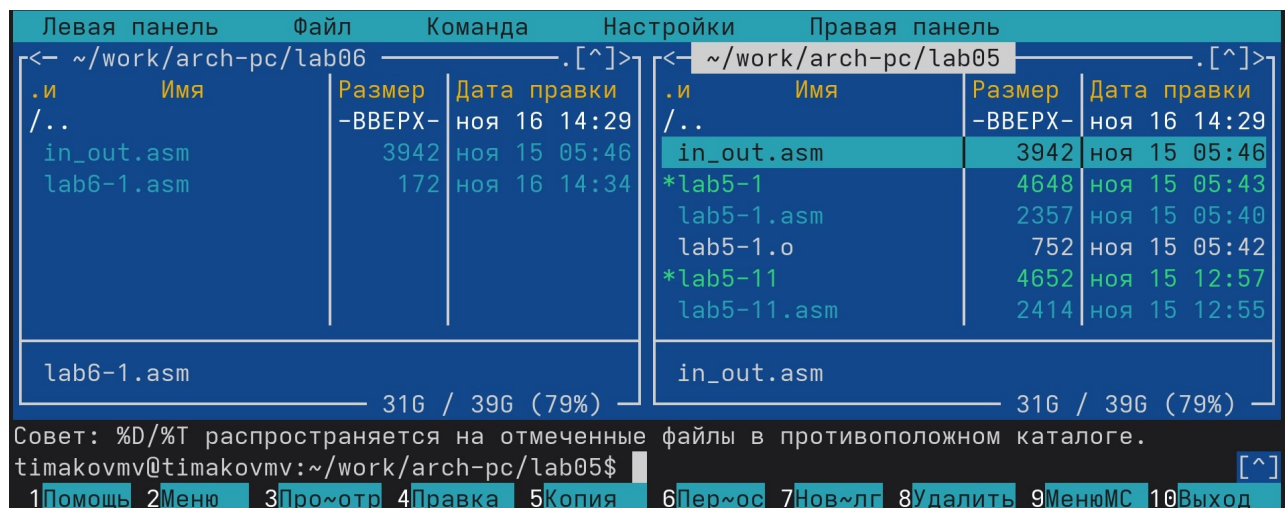


Рисунок 2.3: Копируем файл в Midnight Commander

Создаем исполняемый файл и запускаем его (Рисунок 2.4)

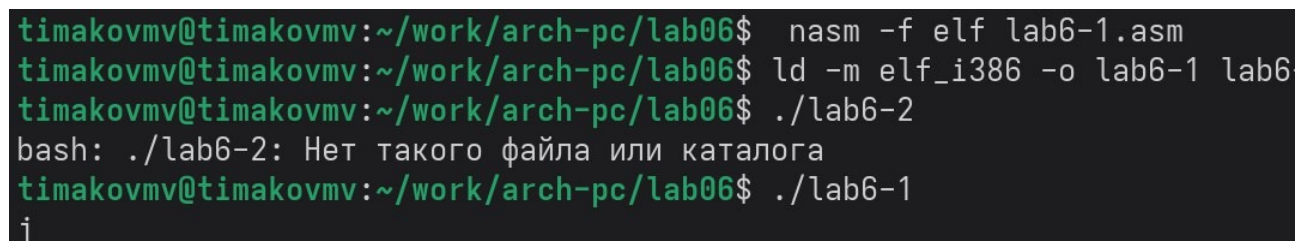


Рисунок 2.4: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убираем кавычки с числовых значений (Рисунок 2.5).

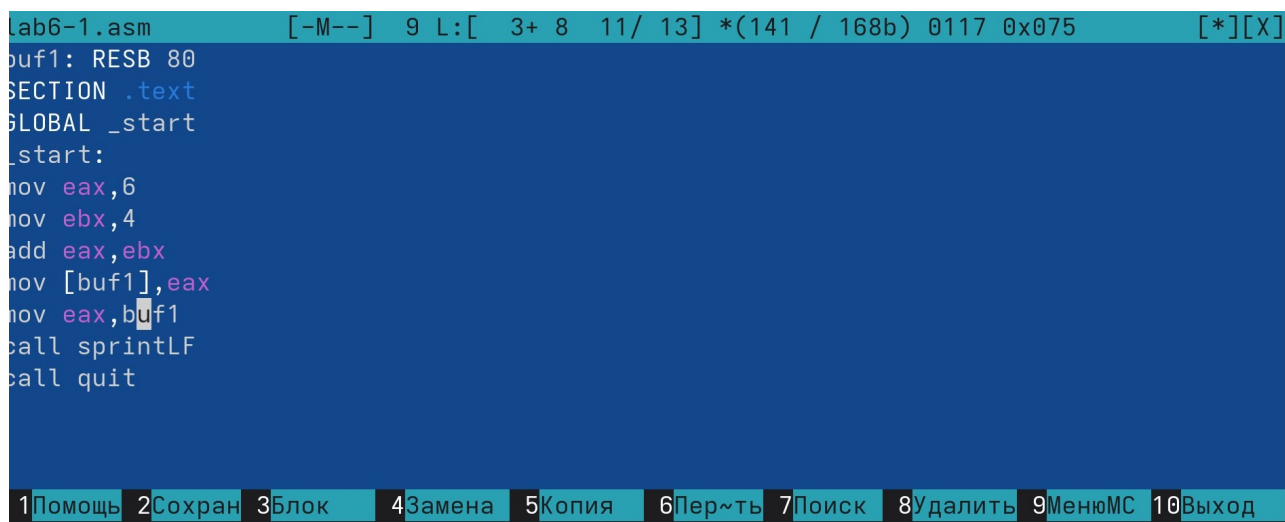


Рисунок 2.5: Изменяем файл

Создаем исполняемый файл и запускаем его (Рисунок 2.6).

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

timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 2.6: Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге (Рисунок 2.7).

```
timakovmv@timakovmv:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o
timakovmv@timakovmv:~/work/arch-pc/lab06$ touch lab6-2.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$ ls
in_out.asm  lab6-1  lab6-1.asm  lab6-1.o  lab6-2.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 2.7: Создаем файл

Заполняем файл в соответствии с листингом 6.2 (Рисунок 2.8).

```
lab6-2.asm  [-M--]  9 L:[  1+ 8  9/  9] *(117 / 117b) <EOF>  [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF
call quit

1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер~ть 7Поиск 8Удалить 9МенюМС 10Выход
```

Рисунок 2.8: Заполняем файл

Создаем исполняемый файл и запускаем его (Рисунок 2.9).

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

Рисунок 2.9: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений (Рисунок 2.10).

```
lab6-2.asm      [-M--]  8 L:[  1+ 4   5/  9] *(66 / 113b) 0054 [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov  eax,6
mov  ebx,4
add  eax,ebx
call iprintLF
call quit
```

1По~щъ 2Со~ан 3Блок 4За~на 5Копия 6Пе~ть 7Поиск 8Уд~ть 9МенюМС10Выход

Рисунок 2.10: Изменяем файл

Создаем исполняемый файл и запускаем его (Рисунок 2.11).

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

Рисунок 2.11: Смотрим на работу программы

Снова открываем файл для редактирования и меняем iprintLF на iprint (Рисунок 2.12).

```
lab6-2.asm      [-M--] 11 L:[  1+ 7   8/  9] *(101 / 111b) 0010 [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:
mov  eax,6
mov  ebx,4
add  eax,ebx
call iprint
call quit
```

1По~щъ 2Со~ан 3Блок 4За~на 5Копия 6Пе~ть 7Поиск 8Уд~ть 9МенюМС10Выход

Рисунок 2.12: Изменяем файл

Создаем исполняемый файл и запускаем его (Рисунок 2.13)

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

Рисунок 2.13: Смотрим на работу программы

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

2.2 Выполнение арифметических операций в NASM

Создаем новый файл в каталоге (Рисунок 2.14).

```
10timakovmv@timakovmv:~/work/arch-pc/lab06$ touch lab6-3.asm  
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 2.14: Создаем файл командой `touch`

Открываем файл и редактируем в соответствии с листингом 6.3 (Рисунок 2.15).

```
-----  
Программа вычисления выражения  
-----  
%include 'in_out.asm' ; подключение внешнего файла  
SECTION .data  
div: DB 'Результат: ',0  
rem: DB 'Остаток от деления: ',0  
66 Демидова А. В.  
Архитектура 386  
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
```

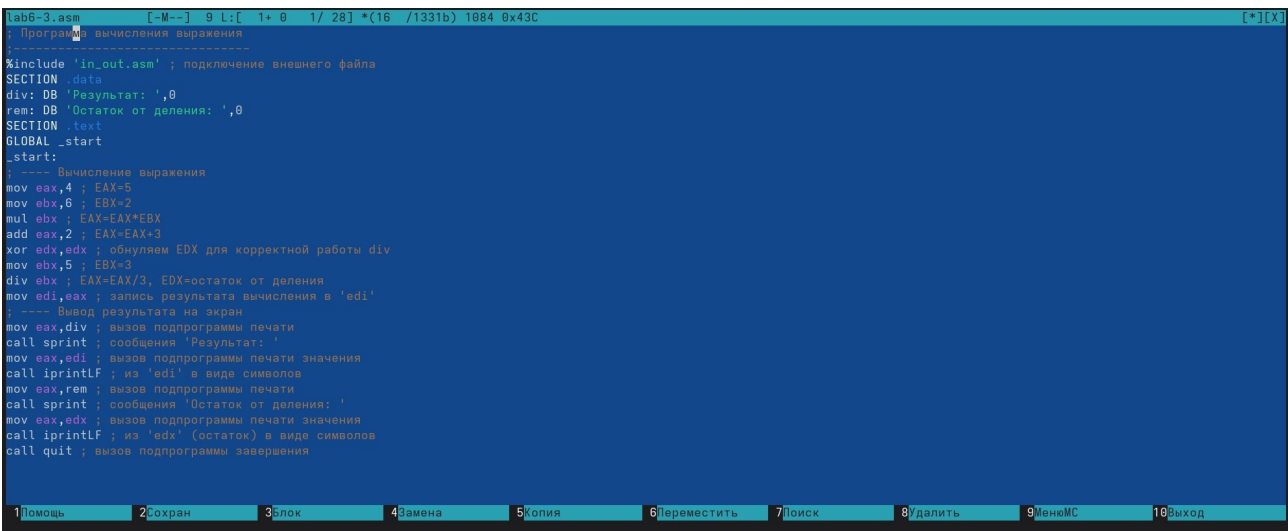
Рисунок 2.15: Заполняем файл

Создаем исполняемый файл и запускаем его (Рисунок 2.16).

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

Рисунок 2.16: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (Рисунок 2.17).



```
lab6-3.asm [W--] 9 L: [ 1+ 0 1/ 28] *(16 /1331b) 1084 0x43C [X]
; Программа: вычисления выражения
;-----
%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=5
mov ebx,6 ; EBX=2
mul ebx ; EAX=EAX*EBX
add eax,2 ; EAX=EAX+2
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

1Помощь 2Сохран 3Скопировать 4Замена 5Скопировать 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рисунок 2.17: Редактируем файл

Компилируем файл и запускаем программу (Рисунок 2.18).

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

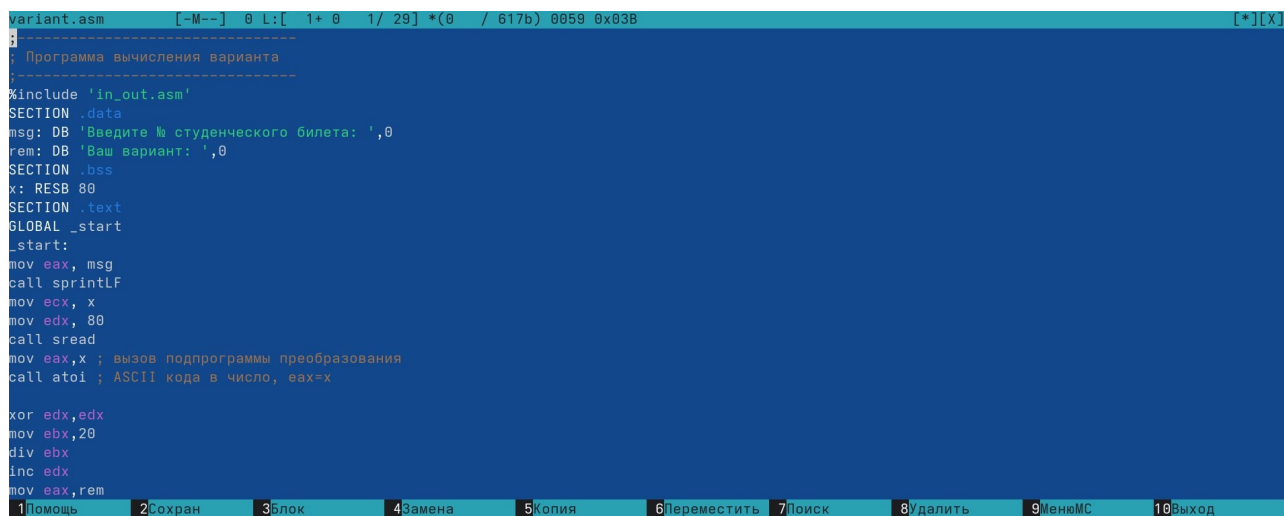
Рисунок 2.18: Рис. 3.17: Смотрим на результат работы программы

Создаем новый файл в каталоге (Рисунок 2.19).

```
timakovmv@timakovmv:~/work/arch-pc/lab06$ touch variant.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 2.19: Создаем файл командой touch

Открываем файл и редактируем в соответствии с листингом 6.4 (Рисунок 2.20).



```
variant.asm [-M--] 0 L:[ 1+ 0 1/ 29] *(0 / 617b) 0059 0x03B [*][X]
;
; 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 read
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
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход
```

Рисунок 2.20: Заполняем файл

Компилируем файл и запускаем его (Рисунок 2.21).

```
timakovmv@timakovmv:~/work/arch-pc/lab06$ nasm -f elf variant.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
timakovmv@timakovmv:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032252478
Ваш вариант: 19
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 2.21: Смотрим на результат работы программы

2.3 Ответы на вопросы

- 1) Строка «mov eax, get» и строка «call sprint»
- 2) Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки.
- 3) Инструкция «call atoi» используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в регистре eax.
- 4) Строка «xor edx, edx» обнуляет регистр edx перед выполнением деления. Строка «mov ebx, 20» загружает значение 20 в регистр ebx. Строка «div ebx» выполняет деление регистра eax на значение регистра ebx с сохранением частного в регистре eax и остатка в регистре edx.
- 5) Остаток от деления записывается в регистр edx.
- 6) Инструкция «inc edx» используется для увеличения значения в регистре edx на 1. В данном случае, она увеличивает остаток от деления на 1.
- 7) Строка «mov eax, edx» передает значение остатка от деления в регистр eax. Строка «call iprintLF» вызывает процедуру iprintLF для вывода значения на экран вместе с переводом строки.

3 Задание для самостоятельной работы

Создаем новый файл в каталоге (Рисунок 3.1).

```
timakovmv@timakovmv:~/work/arch-pc/lab06$ touch lab6-4.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 3.1: Создаем файл

Открываем его и заполняем, чтобы решалось выражение $f(x) = 18 * (x + 1) / 6$ (Рисунок 3.2)

```
/home/timakovmv/work/arch-pc/lab06/lab6-5.asm 519/519 100%
#include "in_out.asm"

SECTION .data
msg: DB "Введите x: ",0
div: DB "Результат: ",0

SECTION .bss
rez: RESB 80
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax,msg
call sprintf

mov ecx,x
mov edx,80
call cread
mov eax,x
call atoi

; Вычисление (1/3 * x + 5) * 7
; 1/3 * x = x / 3
mov ebx,3
cdq
div ebx      ; eax = x / 3
add eax,5    ; eax = x/3 + 5
mov ebx,7
imul eax,ebx ; eax = (x/3 + 5) * 7

mov [rez],eax
mov eax,div
call sprintf
mov eax,[rez]
call iprintf

call quit

1 Помощь 2 Разверн 3 Выход 4 Нех 5 Перейти 6 7 Поиск 8 Исходный 9 Формат 10 Выход
```

Рисунок 3.2: Заполняем файл

Компилируем программу и проверяем для $x=9$ (Рисунок 3.3).

```
timakovmv@timakovmv:~/work/arch-pc/lab06$ nasm -f elf lab6-5.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-5 lab6-5.o
timakovmv@timakovmv:~/work/arch-pc/lab06$ ./lab6-5
Введите x:
9
Результат: 56
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 3.3: Проверяем работу программы

проверяем для $x=3$ (Рисунок 3.4).

```
timakovmv@timakovmv:~/work/arch-pc/lab06$ nasm -f elf lab6-5.asm
timakovmv@timakovmv:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-5 lab6-5.o
timakovmv@timakovmv:~/work/arch-pc/lab06$ ./lab6-5
Введите x:
3
Результат: 42
timakovmv@timakovmv:~/work/arch-pc/lab06$
```

Рисунок 3.4: Проверяем работу программы

4 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.