РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 06__

дисциплина: Архитектура компьютера

Студент: Добрынин Н. А.

Группа: НБИбд-01-25

МОСКВА

2025_ г.

1) Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

2) Самостоятельная работа

Написать программу вычисления выражения $y = f(x)$. Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции $f(x) = (1/3\ x + 5) \cdot 7$ Создать исполняемый файл и проверить его работу для значений $x1$=3 и $x2$=9.

3) Выполнение работы

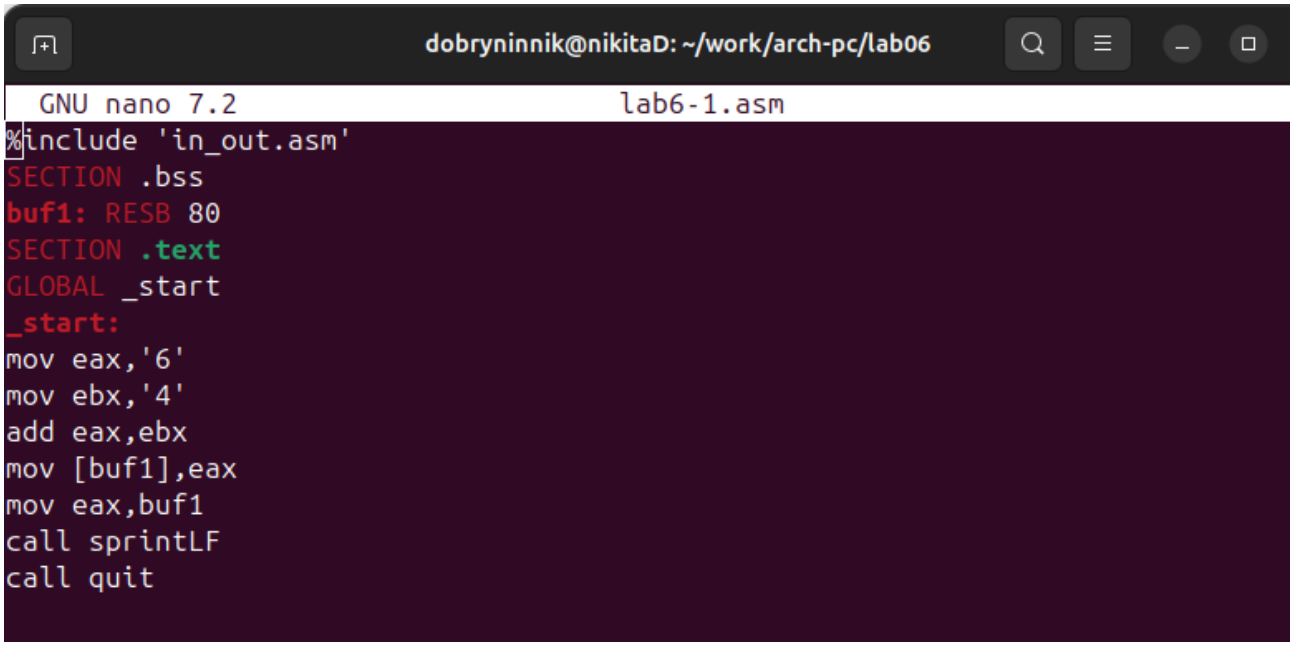

Рис.1 – код программы lab6-1.asm

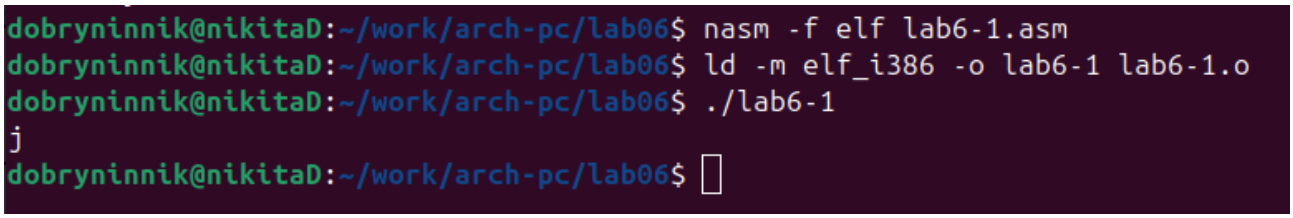

Рис.2 – исполнение программы lab6-1.asm

1) Создал каталог для программ лабораторной работы № 6, перешел в него и создал файл lab6-1.asm. Написал код программы с листинга (Рис.1), скомпилировал и запустил программу, получил вывод j(Рис.2)

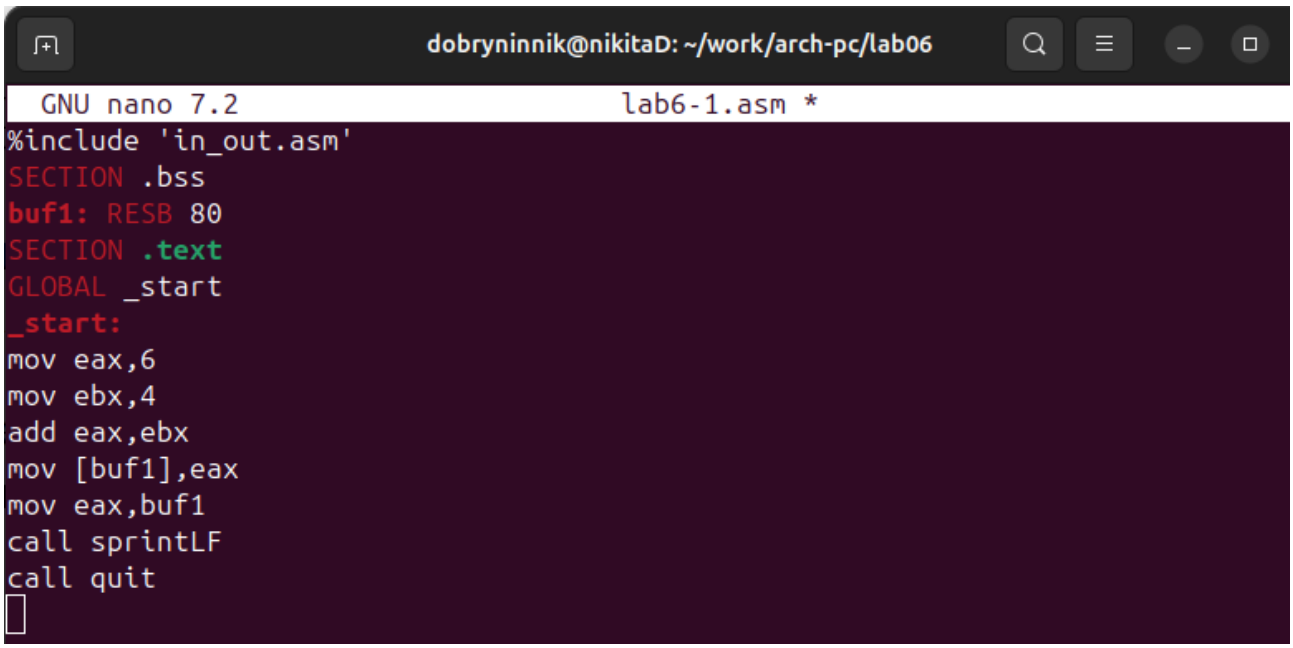

Рис. 3 – исправленный код программы lab6-1.asm

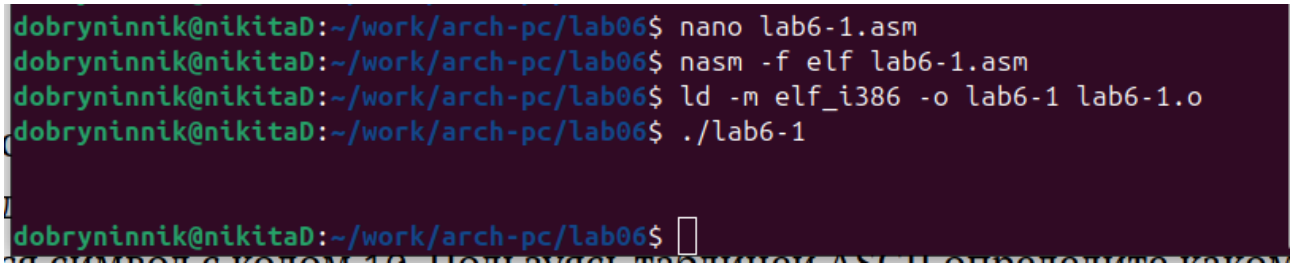

Рис.4 – исполнение исправленного файла lab6-1.asm

2) Исправил код программы lab6-1.asm, а точнее заменил символы на числа(Рис.3). Скомпилировал и запустил исправленную программу lab6-1.asm(Рис.4), никакой символ не виден, но он есть. Это возврат LF

Рис.5 – код программы lab6-2.asm

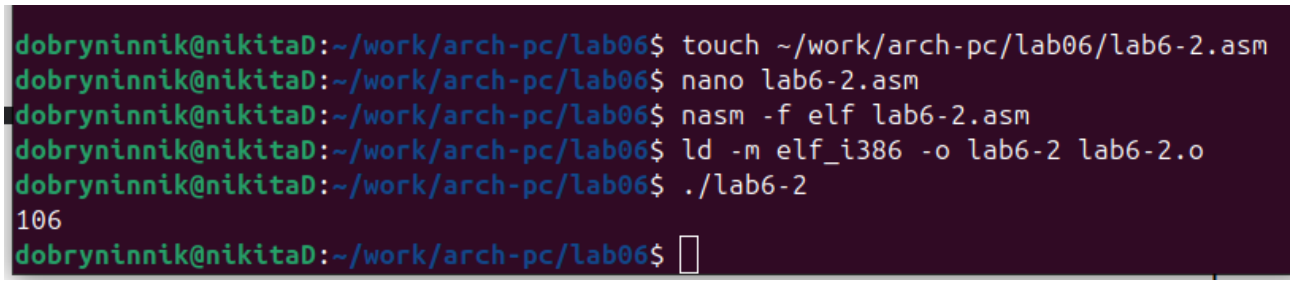

Рис.6 – исполнение кода программы lab6-2.asm

3) Как отмечалось выше, для работы с числами в файле in_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы с использованием этих функций.
   Создал файл lab6-2.asm написал код(Рис.5). Скомпилировал код и запустил его, программа вывела ответ «106»(Рис.6).

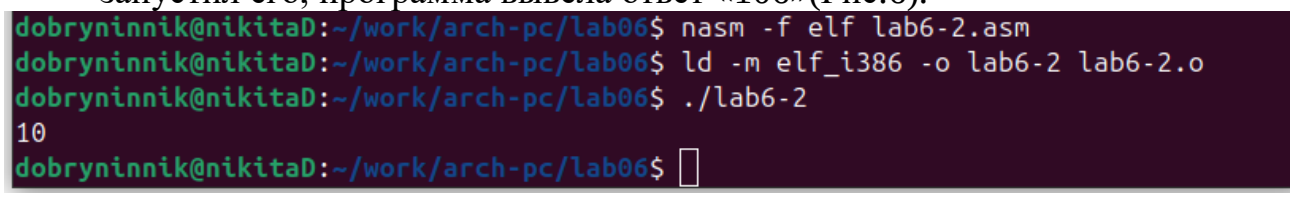

Рис.7 – результат измененного кода программы lab6-2.asm

4) Изменил код программы: сменил символы на числа, теперь программа выводит число «10»(Рис.7)

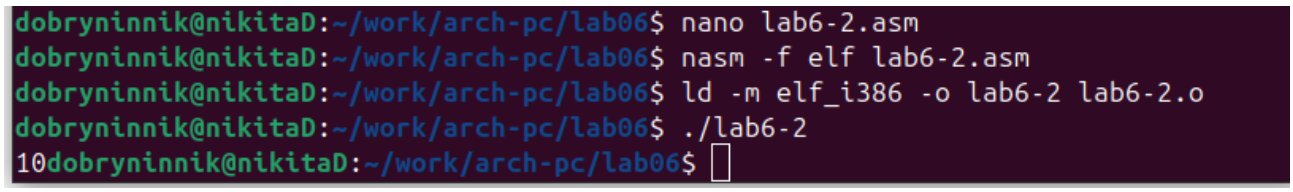

Рис.8 - результат измененного кода программы lab6-2.asm

5) Заменил функцию iprintLF на iprint. Создал исполняемый файл и запустил его(Рис.8). iprintLF и iprint отличаются тем что теперь ответ выводится в

одну строку

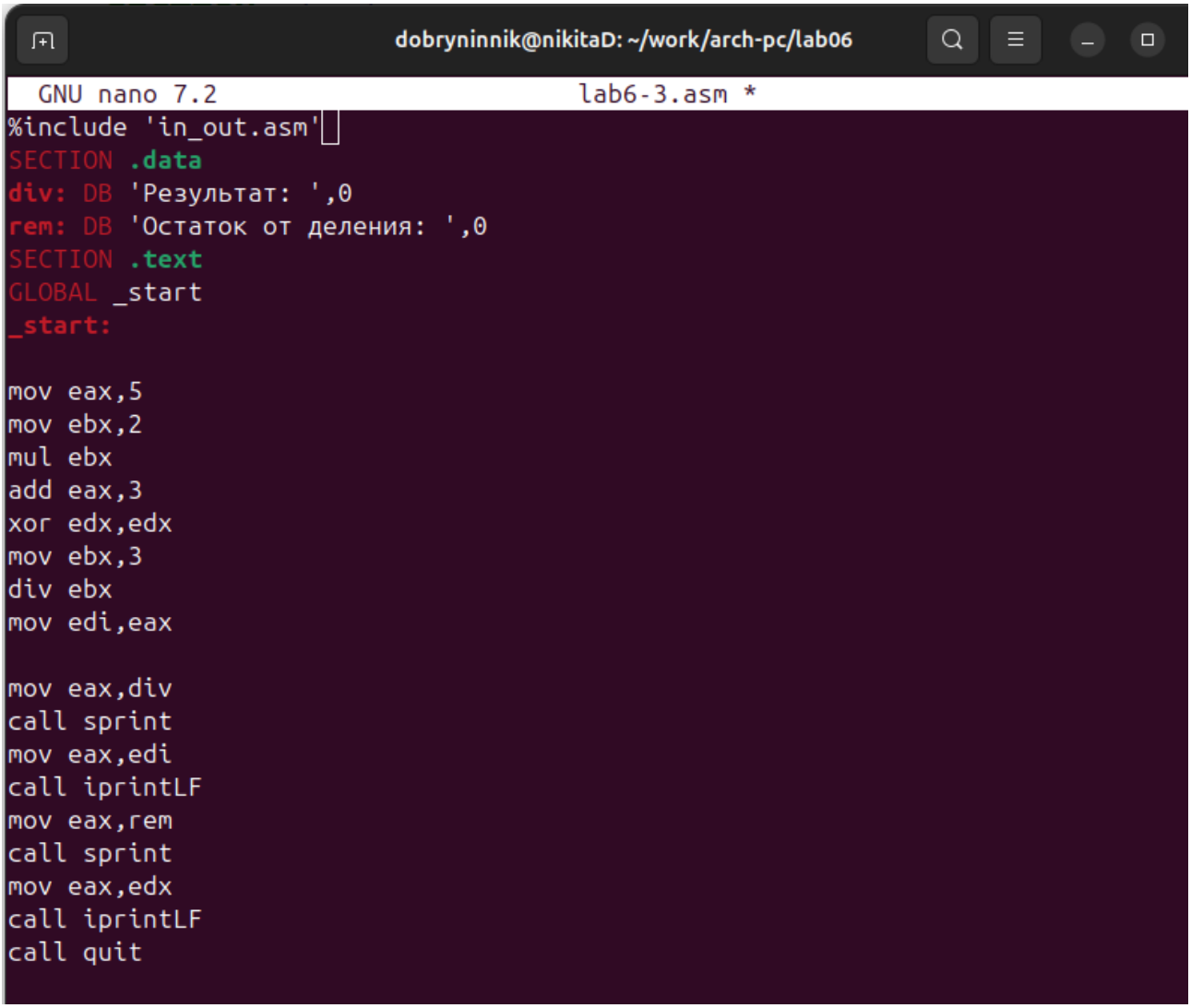

Рис.9 – код программы lab6-3.asm

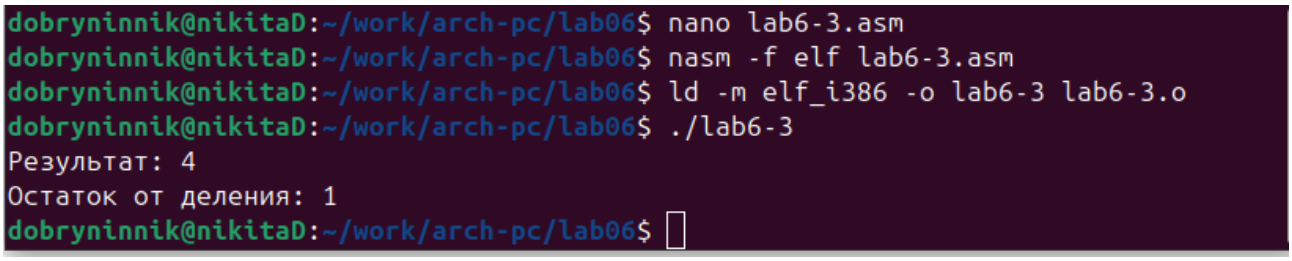

Рис.10 – работа программы lab6-3.asm

6) Выполнил программу вычисления арифметического выражения $f(x) = (5 * 2 + 3)/3$. Написал код программы(Рис.9), скомпилировал и запустил его(Рис.10).

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

Рис.11 – работа измененной программы lab6-3.asm

7) Результат изменённого кода программы lab6-3.asm для вычисления выражения $f(x) = (4 * 6 + 2)/5$ (Рис.11).

dobryninnik@nikitaD: ~/work/arch-pc/lab06

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

Рис.12 – код программы variant.asm

```
dobryninnik@nikitaD:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
dobryninnik@nikitaD:~/work/arch-pc/lab06$ nano variant.asm
dobryninnik@nikitaD:~/work/arch-pc/lab06$ nasm -f elf variant.asm
dobryninnik@nikitaD:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
dobryninnik@nikitaD:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1132255598
Ваш вариант: 19
dobryninnik@nikitaD:~/work/arch-pc/lab06$
```

Рис.13 – исполнение программы variant.asm

8) Рассмотрел программу вычисления варианта задания по номеру студенческого билета, написал код программы(Рис.12), скомпилировал и запустил код, получил номер 19, программа работает корректно(Рис.13)

4) Выполнение самостоятельной работы

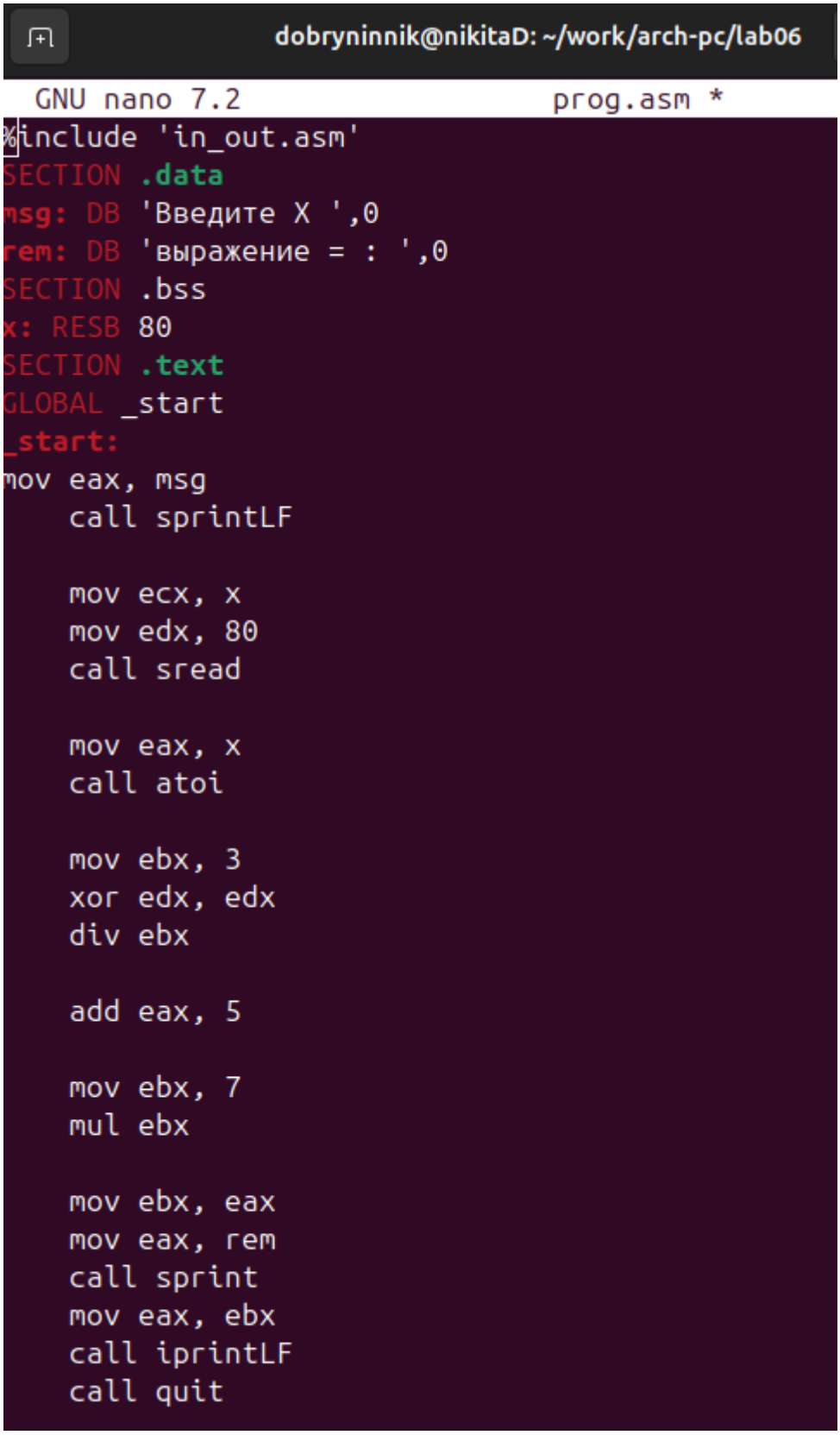

Рис.14 – код программы prog.asm для вычисления выражения f(x)=(1/3 x + 5) · 7

```
dobryninnik@nikitaD:~/work/arch-pc/lab06$ nano prog.asm
dobryninnik@nikitaD:~/work/arch-pc/lab06$ nasm -f elf32 prog.asm
dobryninnik@nikitaD:~/work/arch-pc/lab06$ ld -m elf_i386 -o prog prog.o
dobryninnik@nikitaD:~/work/arch-pc/lab06$ ./prog
Введите X
3
выражение = : 42
dobryninnik@nikitaD:~/work/arch-pc/lab06$ ./prog
Введите X
9
выражение = : 56
dobryninnik@nikitaD:~/work/arch-pc/lab06$
```

Рис.15 – исполнение кода программы prog.asm для x1=3 и x2=9

9) Необходимо написать код вычисления функции под тем номером который выдала программа varian.asm и проверить корректность этой программы для указанных «x». Мне попалось число 19 по этому я беру выражение $f(x) = (1/3\ x + 5) \cdot 7$ при проверке x1=3 и x2=9.
Создал файл prog.asm, написал код вычисляющий нужное выражение(Рис.14), скомпелировал и запустил его(Рис.15), программа при x1=3 выводит ответ 42, а при x2=9 выводит ответ 56, программа работает корректно.

5) Ответы на вопросы

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

В строке mov eax,rem значение переменной с фразой ‘Ваш вариант:’ перекладывается в регистр eax. Строка call sprint вызывает подпрограмму для вывода строки.

2) Для чего используется следующие инструкции?
mov ecx, x
mov edx, 80
call sread

mov ecx, x - перемещает значение переменной X в регистр ecx.
mov edx, 80 - устанавливает значение 80 в регистр edx.
call sread - вызывает подпрограмму для чтения значения с консоли.

3) Для чего используется инструкция “call atoi”?

Эта инструкция вызывает подпрограмму, которая преобразует введенные символы в числовой формат.

4) Какие строки листинга 6.4 отвечают за вычисления варианта?

xor edx,edx - обнуляет регистр edx.
mov ebx,20 - устанавливает значение 20 в регистр ebx.
div ebx - производит деление номера студенческого билета на 20.
inc edx - увеличивает значение регистра edx на 1.

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

Остаток от деления записывается в регистр edx.

6) Для чего используется инструкция “inc edx”?

Инструкция inc edx увеличивает значение регистра edx на 1. В данном случае, она используется для выполнения формулы вычисления варианта, где требуется добавить 1 к остатку от деления.

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

mov eax,edx - результат вычислений перекладывается в регистр eax.
call iprintLF - вызывается подпрограмма для вывода результата на экран.

Заключение

Я изучил работу с арифметическими операциями в NASM